

Published in final edited form as:

Math Program. 2014 February 1; 143(1-2): 339–356. doi:10.1007/s10107-012-0612-1.

MM Algorithms for Geometric and Signomial Programming

Kenneth Lange and

Departments of Biomathematics, Human Genetics, and Statistics, University of California, Los Angeles, CA 90095-1766, USA. klange@ucla.edu

Hua Zhou

Department of Statistics, North Carolina State University, 2311 Stinson Drive, Campus Box 8203, Raleigh, NC 27695-8203, USA. hua_zhou@ncsu.edu

Abstract

This paper derives new algorithms for signomial programming, a generalization of geometric programming. The algorithms are based on a generic principle for optimization called the MM algorithm. In this setting, one can apply the geometric-arithmetic mean inequality and a supporting hyperplane inequality to create a surrogate function with parameters separated. Thus, unconstrained signomial programming reduces to a sequence of one-dimensional minimization problems. Simple examples demonstrate that the MM algorithm derived can converge to a boundary point or to one point of a continuum of minimum points. Conditions under which the minimum point is unique or occurs in the interior of parameter space are proved for geometric programming. Convergence to an interior point occurs at a linear rate. Finally, the MM framework easily accommodates equality and inequality constraints of signomial type. For the most important special case, constrained quadratic programming, the MM algorithm involves very simple updates.

Keywords

arithmetic-geometric mean inequality; geometric programming; global convergence; MM algorithm; linearly constrained quadratic programming; parameter separation; penalty method; signomial programming

1 Introduction

As a branch of convex optimization theory, geometric programming is next in line to linear and quadratic programming in importance [4,5,15,16]. It has applications in chemical equilibrium problems [14], structural mechanics [5], integrated circuit design [7], maximum likelihood estimation [12], stochastic processes [6], and a host of other subjects [5]. Geometric programming deals with posynomials, which are functions of the form

$$f(\mathbf{x}) = \sum_{\alpha \in S} c_{\alpha} \prod_{i=1}^n x_i^{\alpha_i}. \quad (1)$$

Here the index set $S \subset \mathbb{R}^n$ is finite, and all coefficients c_{α} and all components x_1, \dots, x_n of the argument \mathbf{x} of $f(\mathbf{x})$ are positive. The possibly fractional powers α_i corresponding to a particular α may be positive, negative, or zero. For instance, $x_1^{-1} + 2x_1^3 x_2^{-2}$ is a posynomial on \mathbb{R}^2 . In geometric programming we minimize a posynomial $f(\mathbf{x})$ subject to posynomial inequality constraints of the form $u_j(\mathbf{x}) \leq 1$ for $1 \leq j \leq q$, where the $u_j(\mathbf{x})$ are again posynomials. In some versions of geometric programming, equality constraints of posynomial type are permitted [3].

A signomial function has the same form as the posynomial (1), but the coefficients c_{α} are allowed to be negative. A signomial program is a generalization of a geometric program, where the objective and constraint functions can be signomials. From a computational point of view, signomial programming problems are significantly harder to solve than geometric programming problems. After suitable change of variables, a geometric program can be transformed into a convex optimization problem and globally solved by standard methods. In contrast, signomials may have many local minima. Wang et al. [20] recently derived a path algorithm for solving unconstrained signomial programs.

The theory and practice of geometric programming has been stable for a generation, so it is hard to imagine saying anything novel about either. The attractions of geometric programming include its beautiful duality theory and its connections with the arithmetic-geometric mean inequality. The present paper derives new algorithms for both geometric and signomial programming based on a generic device for iterative optimization called the MM algorithm [9,11]. The MM perspective possesses several advantages. First it provides a unified framework for solving both geometric and signomial programs. The algorithms derived here operate by separating parameters and reducing minimization of the objective function to a sequence of one-dimensional minimization problems. Separation of parameters is apt to be an advantage in high-dimensional problems. Another advantage is ease of implementation compared to competing methods of unconstrained geometric and signomial programming [20]. Finally, straightforward generalizations of our MM algorithms extend beyond signomial programming.

We conclude this introduction by sketching a roadmap to the rest of the paper. Section 2 reviews the MM algorithm. Section 3 derives MM algorithm for unconstrained signomial program from two simple inequalities. The behavior of the MM algorithm is illustrated on a few numerical examples in Section 4. Section 5 extends the MM algorithm for unconstrained problems to the constrained cases using the penalty method. Section 6 specializes to linearly constrained quadratic programming on the positive orthant. Convergence results are discussed in Section 7.

2 Background on the MM Algorithm

The MM principle involves majorizing the objective function $f(\mathbf{x})$ by a surrogate function $g(\mathbf{x} | \mathbf{x}_m)$ around the current iterate \mathbf{x}_m (with i th component x_{mi}) of a search. Majorization is defined by the two conditions

$$f(\mathbf{x}_m) = g(\mathbf{x}_m | \mathbf{x}_m) \quad (2)$$

$$f(\mathbf{x}) \leq g(\mathbf{x} | \mathbf{x}_m), \mathbf{x} \neq \mathbf{x}_m.$$

In other words, the surface $\mathbf{x} \mapsto g(\mathbf{x} | \mathbf{x}_m)$ lies above the surface $\mathbf{x} \mapsto f(\mathbf{x})$ and is tangent to it at the point $\mathbf{x} = \mathbf{x}_m$. Construction of the majorizing function $g(\mathbf{x} | \mathbf{x}_m)$ constitutes the first M of the MM algorithm.

The second M of the algorithm minimizes the surrogate $g(\mathbf{x} | \mathbf{x}_m)$ rather than $f(\mathbf{x})$. If \mathbf{x}_{m+1} denotes the minimizer of $g(\mathbf{x} | \mathbf{x}_m)$, then this action forces the descent property $f(\mathbf{x}_{m+1}) < f(\mathbf{x}_m)$. This fact follows from the inequalities

$$f(\mathbf{x}_{m+1}) \leq g(\mathbf{x}_{m+1} | \mathbf{x}_m) \leq g(\mathbf{x}_m | \mathbf{x}_m) = f(\mathbf{x}_m),$$

reflecting the definition of \mathbf{x}_{m+1} and the tangency conditions (2). The descent property makes the MM algorithm remarkably stable. Strictly speaking, the validity of the descent property depends only on decreasing $g(\mathbf{x} | \mathbf{x}_m)$, not on minimizing $g(\mathbf{x} | \mathbf{x}_m)$.

3 Unconstrained Signomial Programming

The art in devising an MM algorithm revolves around intelligent choice of the majorizing function. For signomial programming problems, fortunately one can invoke two simple inequalities. For terms with positive coefficients c_{α} , we use the arithmetic-geometric mean inequality

$$\prod_{i=1}^n z_i^{\alpha_i} \leq \sum_{i=1}^n \frac{\alpha_i}{\|\alpha\|_1} z_i^{\|\alpha\|_1} \quad (3)$$

for nonnegative numbers z_i and α_i and ℓ_1 norm $\|\alpha\|_1 = \sum_{i=1}^n |\alpha_i|$ [19]. If we make the choice $z_i = x_i/x_{mi}$ in inequality (3), then the majorization

$$\prod_{i=1}^n x_i^{\alpha_i} \leq \left(\prod_{i=1}^n x_{mi}^{\alpha_i} \right) \sum_{i=1}^n \frac{\alpha_i}{\|\alpha\|_1} \left(\frac{x_i}{x_{mi}} \right)^{\|\alpha\|_1}, \quad (4)$$

emerges, with equality when $\mathbf{x} = \mathbf{x}_m$. We can broaden the scope of the majorization (4) to cases with $\alpha_i < 0$ by replacing z_i by the reciprocal ratio x_{mi}/x_i whenever $\alpha_i < 0$. Thus, for terms $c_{\alpha} \prod_{i=1}^n x_i^{\alpha_i}$ with $c_{\alpha} > 0$, we have the majorization

$$c_{\alpha} \prod_{i=1}^n x_i^{\alpha_i} \leq c_{\alpha} \left(\prod_{j=1}^n x_{mj}^{\alpha_j} \right) \sum_{i=1}^n \frac{|\alpha_i|}{\|\alpha\|_1} \left(\frac{x_i}{x_{mi}} \right)^{\|\alpha\|_1 \text{sgn}(\alpha_i)},$$

where $\text{sgn}(\alpha_i)$ is the sign function.

The terms $c_{\alpha} \prod_{i=1}^n x_i^{\alpha_i}$ with $c_{\alpha} < 0$ are handled by a different majorization. Our point of departure is the supporting hyperplane minorization

$$z \geq 1 + \ln z$$

at the point $z = 1$. If we let $z = \prod_{i=1}^n (x_i/x_{mi})^{\alpha_i}$, then it follows that

$$\prod_{i=1}^n x_i^{\alpha_i} \geq \prod_{j=1}^n x_{mj}^{\alpha_j} \left(1 + \sum_{i=1}^n \alpha_i \ln x_i - \sum_{i=1}^n \alpha_i \ln x_{mi} \right) \quad (5)$$

is a valid minorization in \mathbf{x} around the point \mathbf{x}_m . Multiplication by the negative coefficient c_{α} now gives the desired majorization. The surrogate function separates parameters and is convex when all of the α_i are positive.

In summary, the objective function (1) is majorized up to an irrelevant additive constant by the sum

$$g(\mathbf{x} | \mathbf{x}_m) = \sum_{i=1}^n g_i(x_i | \mathbf{x}_m)$$

$$g_i(x_i | \mathbf{x}_m) = \sum_{\alpha \in S_+} c_\alpha \left(\prod_{j=1}^n x_{mj}^{\alpha_j} \right) \frac{|\alpha_i|}{\|\alpha\|_1} \left(\frac{x_i}{x_{mi}} \right)^{\|\alpha\|_1 \operatorname{sgn}(\alpha_i)} + \sum_{\alpha \in S_-} c_\alpha \left(\prod_{j=1}^n x_{mj}^{\alpha_j} \right) \alpha_i \ln x_i, \quad (6)$$

where $S_+ = \{\alpha : c_\alpha > 0\}$, and $S_- = \{\alpha : c_\alpha < 0\}$. To guarantee that the next iterate is well defined and occurs on the interior of the parameter domain, it is helpful to assume for each i that at least one $\alpha \in S_+$ has α_i positive and at least one $\alpha \in S_+$ has α_i negative. Under these conditions each $g_i(x_i | \mathbf{x}_m)$ is coercive and attains its minimum on the open interval $(0, \infty)$.

Minimization of the majorizing function is straightforward because the surrogate functions $g_i(x_i | \mathbf{x}_m)$ are univariate functions. The derivative of $g_i(x_i | \mathbf{x}_m)$ with respect to its left argument equals

$$g'_i(x_i | \mathbf{x}_m) = \sum_{\alpha \in S_+} c_\alpha \left(\prod_{j=1}^n x_{mj}^{\alpha_j} \right) \alpha_i x_i^{-1} \left(\frac{x_i}{x_{mi}} \right)^{\|\alpha\|_1 \operatorname{sgn}(\alpha_i)} + \sum_{\alpha \in S_-} c_\alpha \left(\prod_{j=1}^n x_{mj}^{\alpha_j} \right) \alpha_i x_i^{-1}$$

Assuming that the exponents α_i are integers, this is a rational function of x_i , and once we equate it to 0, we are faced with solving a polynomial equation. This task can be accomplished by bisection or by Newton's method. In practice, just a few steps of either algorithm suffice since the MM principle merely requires decreasing the surrogate functions $g_i(x_i | \mathbf{x}_m)$.

In a geometric program, the function $g'_i(x_i | \mathbf{x}_m)$ has a single root on the interval $(0, \infty)$. For a proof of this fact, note that making the standard change of variables $x_i = e^{y_i}$ eliminates the positivity constraint $x_i > 0$ and renders the transformed function $h_i(y_i | \mathbf{x}_m) = g_i(x_i | \mathbf{x}_m)$ strictly convex. Because $|\alpha_i| \operatorname{sgn}(\alpha_i)^2 = |\alpha_i|$, the second derivative

$$h''_i(y_i | \mathbf{x}_m) = \sum_{\alpha \in S_+} c_\alpha \left(\prod_{j=1}^n x_{mj}^{\alpha_j} \right) \frac{|\alpha_i| \cdot \|\alpha\|_1}{x_{mi}^{\|\alpha\|_1 \operatorname{sgn}(\alpha_i)}} e^{\|\alpha\|_1 \operatorname{sgn}(\alpha_i) y_i}$$

is positive. Hence, $h_i(y_i | \mathbf{x}_m)$ is strictly convex and possesses a unique minimum point. These arguments yield the even sweeter dividend that the MM iteration map is continuously differentiable. From the vantage point of the implicit function theorem [8], the stationary condition $h'_i(y_{m+1,i} | \mathbf{x}_m) = 0$ determines $y_{m+1,i}$, and consequently $x_{m+1,i}$, in terms of \mathbf{x}_m . Observe here that $h''_i(y_{mi} | \mathbf{x}_m) \neq 0$ as required by the implicit function.

It is also worth pointing out that even more functions can be brought under the umbrella of signomial programming. For instance, majorization of the two related functions $-\ln f(\mathbf{x})$ and $\ln f(\mathbf{x})$ is possible for any posynomial $f(\mathbf{x}) = \sum_{\alpha} c_\alpha \prod_{i=1}^n x_i^{\alpha_i}$. In the first case,

$$-\ln f(\mathbf{x}) \leq -\sum \frac{a_{m\alpha}}{\alpha} \frac{1}{b_m} \left[\sum_{i=1}^n \alpha_i \ln x_i + \ln \left(\frac{c_\alpha b_m}{a_{m\alpha}} \right) \right] \quad (7)$$

holds for $a_{m\alpha} = c_\alpha \prod_{i=1}^n x_{mi}^{\alpha_i}$ and $b_m = \sum_{\alpha} \alpha_{m\alpha}$ because Jensen's inequality applies to the convex function $-\ln t$. In the second case, the supporting hyperplane inequality applied to the convex function $-\ln t$ implies

$$\ln f(\mathbf{x}) \leq \ln f(\mathbf{x}_m) + \frac{1}{f(\mathbf{x}_m)} [f(\mathbf{x}) - f(\mathbf{x}_m)].$$

This puts us back in the position of needing to majorize a posynomial, a problem we have already discussed in detail. By our previous remarks, the coefficients c_α can be negative as well as positive in this case. Similar majorizations apply to any composition $\phi \circ f(\mathbf{x})$ of a posynomial $f(\mathbf{x})$ with an arbitrary concave function $\phi(y)$.

4 Examples of Unconstrained Minimization

Our first examples demonstrate the robustness of the MM algorithms in minimization and illustrate some of the complications that occur. In each case we can explicitly calculate the MM updates. To start, consider the posynomial

$$f_1(\mathbf{x}) = \frac{1}{x_1^3} + \frac{3}{x_1 x_2^2} + x_1 x_2$$

with the implied constraints $x_1 > 0$ and $x_2 > 0$. The majorization (4) applied to the third term of $f_1(\mathbf{x})$ yields

$$x_1 x_2 \leq x_{m1} x_{m2} \left[\frac{1}{2} \left(\frac{x_1}{x_{m1}} \right)^2 + \frac{1}{2} \left(\frac{x_2}{x_{m2}} \right)^2 \right] = \frac{x_{m2}}{2x_{m1}} x_1^2 + \frac{x_{m1}}{2x_{m2}} x_2^2.$$

Applied to the second term of $f_1(\mathbf{x})$ using the reciprocal ratios, it gives

$$\frac{3}{x_1 x_2^2} \leq \frac{3}{x_{m1} x_{m2}^2} \left[\frac{1}{3} \left(\frac{x_{m1}}{x_1} \right)^3 + \frac{2}{3} \left(\frac{x_{m2}}{x_2} \right)^3 \right] = \frac{x_{m1}^2}{x_{m2}^2} \frac{1}{x_1^3} + \frac{2x_{m2}}{x_{m1}} \frac{1}{x_2^3}.$$

The sum $g(\mathbf{x} | \mathbf{x}_m)$ of the two surrogate functions

$$g_1(x_1 | \mathbf{x}_m) = \frac{1}{x_1^3} + \frac{x_{m1}^2}{x_{m2}^2} \frac{1}{x_1^3} + \frac{x_{m2}}{2x_{m1}} x_1^2$$

$$g_2(x_2 | \mathbf{x}_m) = \frac{2x_{m2}}{x_{m1}} \frac{1}{x_2^3} + \frac{x_{m1}}{2x_{m2}} x_2^2$$

majorizes $f_1(\mathbf{x})$. If we set the derivatives

$$g_1'(x_1 | \mathbf{x}_m) = -\frac{3}{x_1^4} - \frac{x_{m1}^2}{x_{m2}^2} \frac{3}{x_1^4} + \frac{x_{m2}}{x_{m1}} x_1$$

$$g_2'(x_1 | \mathbf{x}_m) = -\frac{6x_{m2}}{x_{m1}} \frac{1}{x_2^4} + \frac{x_{m1}}{x_{m2}} x_2$$

of each of these equal to 0, then the updates

$$x_{m+1,1} = \sqrt[5]{3 \left(\frac{x_{m1}^2}{x_{m2}^2} + 1 \right) \frac{x_{m1}}{x_{m2}}}, x_{m+1,2} = \sqrt[5]{6 \frac{x_{m2}^2}{x_{m1}^2}}$$

solve the minimization step of the MM algorithm. It is also obvious that the point $\mathbf{x} = (\sqrt[5]{6}, \sqrt[5]{6})^t$ is a fixed point of the updates, and the reader can check that it minimizes $f_1(\mathbf{x})$.

It is instructive to consider the slight variations

$$f_2(\mathbf{x}) = \frac{1}{x_1 x_2^2} + x_1 x_2^2$$

$$f_3(\mathbf{x}) = \frac{1}{x_1 x_2^2} + x_1 x_2$$

of this objective function. In the first case, the reader can check that the MM algorithm iterates according to

$$x_{m+1,1} = \sqrt[3]{\frac{x_{m1}^2}{x_{m2}^2}}, x_{m+1,2} = \sqrt[3]{\frac{x_{m2}}{x_{m1}}}$$

In the second case, it iterates according to

$$x_{m+1,1} = \sqrt[5]{\frac{x_{m1}^3}{x_{m2}^3}}, x_{m+1,2} = \sqrt[5]{2 \frac{x_{m2}^2}{x_{m1}^2}}$$

The objective function $f_2(\mathbf{x})$ attains its minimum value whenever $x_1 x_2^2 = 1$. The MM algorithm for $f_2(\mathbf{x})$ converges after a single iteration to the value 2, but the converged point depends on the initial point \mathbf{x}_0 . The infimum of $f_3(\mathbf{x})$ is 0. This value is attained

asymptotically by the MM algorithm, which satisfies the identities $x_{m1} x_{m2}^{3/2} = 2^{3/10}$ and $x_{m+1,2} = 2^{2/25} x_{m2}$ for all $m \geq 1$. These results imply that x_{m1} tends to 0 and x_{m2} to ∞ in such a manner that $f_3(\mathbf{x}_m)$ tends to 0. One could not hope for much better behavior of the MM algorithm in these two examples.

The function

$$f_4(\mathbf{x}) = x_1^2 x_2^2 - 2x_1 x_2 x_3 x_4 + x_3^2 x_4^2 = (x_1 x_2 - x_3 x_4)^2$$

is a signomial but not a posynomial. The surrogate function (6) reduces to

$$g(\mathbf{x} | \mathbf{x}_m) = \frac{x_{m2}^2}{2x_{m1}^2} x_1^4 + \frac{x_{m1}^2}{2x_{m2}^2} x_2^4 + \frac{x_{m4}^2}{2x_{m3}^2} x_3^4 + \frac{x_{m3}^2}{2x_{m4}^2} x_4^4 - 2x_{m1} x_{m2} x_{m3} x_{m4} (\ln x_1 + \ln x_2 + \ln x_3 + \ln x_4)$$

with all variables separated. The MM updates

$$x_{m+1,1} = \sqrt[4]{\frac{x_{m1}^3 x_{m3} x_{m4}}{x_{m2}}}, x_{m+1,2} = \sqrt[4]{\frac{x_{m2}^3 x_{m3} x_{m4}}{x_{m1}}}$$

$$x_{m+1,3} = \sqrt[4]{\frac{x_{m3}^3 x_{m1} x_{m2}}{x_{m4}}}, x_{m+1,4} = \sqrt[4]{\frac{x_{m4}^3 x_{m1} x_{m2}}{x_{m3}}}$$

converge in a single iteration to a solution of $f_4(\mathbf{x}) = 0$. Again the limit depends on the initial point.

The function

$$f_5(\mathbf{x}) = x_1 x_2 + x_1 x_3 + x_2 x_3 - \ln(x_1 + x_2 + x_3)$$

is more complicated than a signomial. It also is unbounded because the point $\mathbf{x} = (m, \frac{1}{m}, \frac{1}{m})$ satisfies $f_5(\mathbf{x}) = 2 + m^{-2} - \ln(m + 2/m)$. According to the majorization (7), an appropriate surrogate is

$$\begin{aligned} g(\mathbf{x} | \mathbf{x}_m) &= \left(\frac{x_{m2}}{2x_{m1}} + \frac{x_{m3}}{2x_{m1}} \right) x_1^2 + \left(\frac{x_{m1}}{2x_{m2}} + \frac{x_{m3}}{2x_{m2}} \right) x_2^2 \\ &+ \left(\frac{x_{m1}}{2x_{m3}} + \frac{x_{m2}}{2x_{m3}} \right) x_3^2 \\ &- \frac{x_{m1}}{x_{m1} + x_{m2} + x_{m3}} \ln x_1 \\ &- \frac{x_{m2}}{x_{m1} + x_{m2} + x_{m3}} \ln x_2 \\ &- \frac{x_{m3}}{x_{m1} + x_{m2} + x_{m3}} \ln x_3 \end{aligned}$$

up to an irrelevant constant. The MM updates are

$$x_{m+1,i} = \sqrt{\frac{x_{mi}^2}{\left(\sum_{j \neq i} x_{mj}\right)(x_{m1} + x_{m2} + x_{m3})}}$$

If the components of the initial point coincide, then the iterates converge in a single iteration to the saddle point with all components equal to $1/\sqrt{6}$. Otherwise, it appears that $f_5(\mathbf{x}_m)$ tends to $-\infty$.

The following objective functions

$$f_6(\mathbf{x}) = x_1^2 x_2^6 + x_1^2 x_2^4 - 2x_1^2 x_2^3 - x_1^2 x_2^2 + 5.25x_1 x_2^3 - 2x_1^2 x_2 + 4.5x_1 x_2^2 + 3x_1^2 + 3x_1 x_2 - 12.75x_1$$

$$f_7(\mathbf{x}) = \sum_{i=1}^{10} x_i^4 + 2 \sum_{i=1}^9 x_i^2 \sum_{j=i+1}^{10} x_j^2 + (10^{-5} - 0.5) \sum_{i=1}^{10} x_i^2 - (2 \times 10^{-5}) \sum_{i=7}^{10} x_i + \frac{1}{16}$$

$$f_8(\mathbf{x}) = x_1 x_3^2 x_6^{-1} x_7^{-1} + x_1^2 x_3^{-1} x_5^{-2} x_6^{-1} x_7 + x_1^3 x_2^2 x_5^{-2} x_6^2 + x_2^{-1} x_4^{-1} x_6^2 + x_3 x_5^3 x_6^{-3}$$

$$f_9(\mathbf{x}) = x_1 x_4^2 + x_2 x_3 + x_1 x_2 x_3 x_4^2 + x_1^{-1} x_4^{-2}$$

from the reference [20] are intended for numerical illustration. Table 1 lists initial conditions, minimum points, minimum values, and number of iterations until convergence under the MM algorithm. Convergence is declared when the relative change in the objective function is less than a pre-specified value ε , in other words, when

$$\frac{f(\mathbf{x}_m) - f(\mathbf{x}_{m+1})}{|f(\mathbf{x}_m)| + 1} \leq \varepsilon.$$

Optimization of the univariate surrogate functions easily succumbs to Newton's method. The MM algorithm takes fewer iterations to converge than the path algorithm for all of the test functions mentioned in [20] except $f_6(\mathbf{x})$. Furthermore, the MM algorithm avoids calculation of the gradient and Hessian and requires no matrix decompositions or selection of tuning constants.

As Section 7 observes, MM algorithms typically converge at a linear rate. Although slow convergence can occur for functions such as the test function $f_6(\mathbf{x})$, there are several ways to accelerate an MM algorithm. For example, our published quasi-Newton acceleration [21] often reduces the necessary number of iterations by one or two orders of magnitude. Figure 1 shows the progress of the MM iterates for the test function $f_6(\mathbf{x})$ with and without quasi-Newton acceleration. Under a convergence criterion of $\varepsilon = 10^{-9}$ and $q = 1$ secant condition, the required number of iterations falls to 30; under the same convergence criterion and $q = 2$ secant conditions, the required number of iterations falls to 12. It is also worth emphasizing that separation of parameters enables parallel processing in high-dimensional problems. We have recently argued [25] that the best approach to parallel processing is through graphics processing units (GPUs). These cheap hardware devices offer one to two orders of magnitude acceleration in many MM algorithms with parameters separated.

5 Constrained Signomial Programming

Extending the MM algorithm to constrained geometric and signomial programming is challenging. Box constraints $a_i \leq x_i \leq b_i$ are consistent with parameter separation as just developed, but more complicated posynomial constraints that couple parameters are not. Posynomial inequality constraints take the form

$$h(\mathbf{x}) = \sum_{\beta} d_{\beta} \prod_{i=1}^n x_i^{\beta_i} \leq 1.$$

The corresponding equality constraint sets $h(\mathbf{x}) = 1$. We propose handling both constraints by penalty methods. Before we treat these matters in more depth, let us relax the positivity restrictions on the d_{β} but enforce the restriction $\beta_i \geq 0$. The latter objective can be achieved

by multiplying $h(\mathbf{x})$ by $x_i^{\max\{\beta_i, 0\}}$ for all i . If we subtract the two sides of the resulting equality, then the equality constraint $h(\mathbf{x}) = 1$ can be rephrased as $r(\mathbf{x}) = 0$ for the signomial $r(\mathbf{x}) = \sum_{\gamma} e_{\gamma} \prod_{i=1}^n x_i^{\gamma_i}$, with no restriction on the signs of the e_{γ} but with the requirement $\gamma_i \geq 0$ in effect. For example, the equality constraint

$$\frac{1}{x_1} + \frac{x_1}{x_2^2} = 1$$

becomes

$$x_1^2 + x_2^2 - x_1 x_2^2 = 0.$$

In the quadratic penalty method [1,13,17] with objective function $f(\mathbf{x})$ and a single equality constraint $r(\mathbf{x}) = 0$ and a single inequality constraint $s(\mathbf{x}) \geq 0$, one minimizes the sum

$f_{\lambda}(\mathbf{x}) = f(\mathbf{x}) + \lambda r(\mathbf{x})^2 + \lambda s(\mathbf{x})_+^2$, where $s(\mathbf{x})_+ = \max\{s(\mathbf{x}), 0\}$. As the penalty constant λ tends to ∞ , the solution vector \mathbf{x}_{λ} typically converges to the constrained minimum. In the revised objective function, the term $r(\mathbf{x})^2$ is a signomial whenever $r(\mathbf{x})$ is a signomial. For example, in our toy problem the choice $r(\mathbf{x}) = x_1^2 + x_2^2 - x_1 x_2^2$ has square

$$r(\mathbf{x})^2 = x_1^4 + x_2^4 + x_1^2 x_2^4 + 2x_1^2 x_2^2 - 2x_1 x_2^4 - 2x_1^3 x_2^2.$$

Of course, the powers in $r(\mathbf{x})$ can be fractional here as well as integer. The term $s(\mathbf{x})_+^2$ is not a signomial and must be subjected to the majorization

$$s(\mathbf{x})_+^2 \leq \begin{cases} [s(\mathbf{x}) - s(\mathbf{x}_m)]^2 & s(\mathbf{x}_m) < 0 \\ s(\mathbf{x})^2 & s(\mathbf{x}_m) \geq 0 \end{cases}$$

to achieve this status. In practice, one does not need to fully minimize $f_{\lambda}(\mathbf{x})$ for any fixed λ . If one increases λ slowly enough, then it usually suffices to merely decrease $f_{\lambda}(\mathbf{x})$ at each iteration. The MM algorithm is designed to achieve precisely this goal. Our exposition so far suggests that we majorize $r(\mathbf{x})^2$, $s(\mathbf{x})^2$, and $[s(\mathbf{x}) - s(\mathbf{x}_m)]^2$ in exactly the same manner that we majorize $f(\mathbf{x})$. Separation of parameters generalizes, and the resulting MM algorithm

keeps all parameters positive while permitting pertinent parameters to converge to 0. Section 7 summarizes some of the convergence properties of this hybrid procedure.

The quadratic penalty method traditionally relies on Newton's method to minimize the unconstrained functions $f_\lambda(\mathbf{x})$. Unfortunately, this tactic suffers from roundoff errors and numerical instability. Some of these problems disappear with the MM algorithm. No matrix inversions are involved, and iterates enjoy the descent property. Ill-conditioning does cause harm in the form of slow convergence, but the previously mentioned quasi-Newton acceleration largely remedies the situation [21]. As an alternative to quadratic penalties, exact penalties take the form $\lambda|r(\mathbf{x})| + \lambda s(\mathbf{x})_+$. Remarkably, the exact penalty method produces the constrained minimum, not just in the limit, but for all finite λ beyond a certain point. Although this desirable property avoids the numerical instability encountered in the quadratic penalty method, the kinks in the objective functions $f(\mathbf{x}) + \lambda|r(\mathbf{x})| + \lambda s(\mathbf{x})_+$ are a nuisance. Our recent paper [24] on the exact penalty method shows how to circumvent this annoyance.

6 Nonnegative Quadratic Programming

As an illustration of constrained signomial programming, consider quadratic programming over the positive orthant. Let

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^t \mathbf{Q} \mathbf{x} + \mathbf{c}^t \mathbf{x}$$

be the objective function, $\mathbf{E} \mathbf{x} = \mathbf{d}$ the linear equality constraints, and $\mathbf{A} \mathbf{x} \leq \mathbf{b}$ the linear inequality constraints. The symmetric matrix \mathbf{Q} can be negative definite, indefinite, or positive definite. The quadratic penalty method involves minimizing the sequence of penalized objective functions

$$f_\lambda(\mathbf{x}) = \frac{1}{2} \mathbf{x}^t \mathbf{Q} \mathbf{x} + \mathbf{c}^t \mathbf{x} + \frac{\lambda}{2} \|(\mathbf{A} \mathbf{x} - \mathbf{b})_+\|_2^2 + \frac{\lambda}{2} \|\mathbf{E} \mathbf{x} - \mathbf{d}\|_2^2$$

as λ tends to ∞ . Based on the obvious majorization

$$x_+^2 \leq \begin{cases} (x - x_m)^2 & x_m < 0 \\ x^2 & x_m \geq 0 \end{cases},$$

the term $\|(\mathbf{A} \mathbf{x} - \mathbf{b})_+\|_2^2$ is majorized by $\|\mathbf{A} \mathbf{x} - \mathbf{b} - \mathbf{r}_m\|_2^2$, where

$$\mathbf{r}_m = \min\{\mathbf{A} \mathbf{x}_m - \mathbf{b}, \mathbf{0}\}.$$

A brief calculation shows that $f_\lambda(\mathbf{x})$ is majorized by the surrogate function

$$g_\lambda(\mathbf{x} | \mathbf{x}_m) = \frac{1}{2} \mathbf{x}^t \mathbf{H}_\lambda \mathbf{x} + \mathbf{v}_{\lambda m}^t \mathbf{x}$$

up to an irrelevant constant, where \mathbf{H}_λ and $\mathbf{v}_{\lambda m}$ are defined by

$$\mathbf{H}_\lambda = \mathbf{Q} + \lambda(\mathbf{A}^t \mathbf{A} + \mathbf{E}^t \mathbf{E})$$

$$\mathbf{v}_{\lambda m} = \mathbf{c} - \lambda \mathbf{A}^t (\mathbf{b} + \mathbf{r}_m) - \lambda \mathbf{E}^t \mathbf{d}.$$

It is convenient to assume that the diagonal coefficients $\frac{1}{2}h_{\lambda ii}$ appearing in the quadratic form $\frac{1}{2}\mathbf{x}^T \mathbf{H}_\lambda \mathbf{x}$ are positive. This is generally the case for large λ . One can handle the off-diagonal term $h_{\lambda ij}x_i x_j$ by either the majorization (4) or the majorization (5) according to the sign of $h_{\lambda ij}$. The reader can check that the MM updates reduce to

$$x_{m+1,i} = \frac{x_{mi}}{2} \left[-\frac{v_{\lambda mi}}{h_{\lambda mi}^+} + \sqrt{\left(\frac{v_{\lambda mi}}{h_{\lambda mi}^+}\right)^2 - 4\frac{h_{\lambda mi}^-}{h_{\lambda mi}^+}} \right], \quad (8)$$

where

$$h_{\lambda mi}^+ = \sum_{j:h_{\lambda ij}>0} h_{\lambda ij}x_{mj}, \quad h_{\lambda mi}^- = \sum_{j:h_{\lambda ij}<0} h_{\lambda ij}x_{mj}.$$

When $h_{\lambda mi}^- = 0$, the update (8) collapses to

$$x_{m+1,i} = x_{mi} \max \left\{ -\frac{v_{\lambda mi}}{h_{\lambda mi}^+}, 0 \right\}. \quad (9)$$

To avoid sticky boundaries, we replace 0 in equation (9) by a small positive constant ε such as 10^{-9} . Sha et al. [18] derived the update (8) for $\lambda = 0$ ignoring the constraints $\mathbf{E}\mathbf{x} = \mathbf{d}$ and $\mathbf{A}\mathbf{x} = \mathbf{b}$.

For a numerical example without equality constraints take

$$f_{10}(\mathbf{x}) = \frac{1}{2}x_1^2 + x_2^2 - x_1x_2 - 2x_1 - 6x_2$$

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ -1 & 2 \\ 2 & 1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 2 \\ 2 \\ 3 \end{pmatrix}.$$

The minimum occurs at the point $(2/3, 4/3)^t$. Table 2 lists the number of iterations until convergence and the converged point \mathbf{x}_λ for the sequence of penalty constants $\lambda = 2^k$. The quadratic program

$$f_{11}(\mathbf{x}) = -8x_1 - 16x_2 + x_1^2 + 4x_2^2$$

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 4 \\ 3 \end{pmatrix}$$

converges much more slowly. Its minimum occurs at the point $(2.4, 1.6)^t$. Table 3 lists the numbers of iterations until convergence with $(q = 1)$ and without $(q = 0)$ acceleration and the converged point \mathbf{x}_λ for the same sequence of penalty constants $\lambda = 2^k$. Fortunately, quasi-Newton acceleration compensates for ill conditioning in this test problem.

7 Convergence

As we have seen, the behavior of the MM algorithm is intimately tied to the behavior of the objective function $f(\mathbf{x})$. For the sake of simplicity, we now restrict attention to unconstrained minimization of posynomials and investigate conditions guaranteeing that $f(\mathbf{x})$ possesses a unique minimum on its domain. Uniqueness is related to the strict convexity of the reparameterization

$$h(\mathbf{y}) = \sum_{\alpha \in S} c_\alpha e^{\alpha^t \mathbf{y}}$$

of $f(\mathbf{x})$, where $\alpha^t \mathbf{y} = \sum_{i=1}^n \alpha_i y_i$ is the inner product of α and \mathbf{y} and $x_i = e^{y_i}$ for each i . The Hessian matrix

$$d^2 h(\mathbf{y}) = \sum_{\alpha \in S} c_\alpha e^{\alpha^t \mathbf{y}} \alpha \alpha^t$$

of $h(\mathbf{y})$ is positive semidefinite, so $h(\mathbf{y})$ is convex. If we let T be the subspace of \mathbb{R}^n spanned by $\{\alpha\}_{\alpha \in S}$, then $h(\mathbf{y})$ is strictly convex if and only if $T = \mathbb{R}^n$. Indeed, suppose the condition holds. For any $\mathbf{v} \neq \mathbf{0}$, it then must be true that $\alpha^t \mathbf{v} \neq 0$ for some $\alpha \in S$. As a consequence,

$$\mathbf{v}^t d^2 h(\mathbf{y}) \mathbf{v} = \sum_{\alpha \in S} c_\alpha e^{\alpha^t \mathbf{y}} (\alpha^t \mathbf{v})^2 > 0,$$

and $d^2 h(\mathbf{y})$ is positive definite. Conversely, suppose $T \neq \mathbb{R}^n$, and take $\mathbf{v} \neq \mathbf{0}$ with $\alpha^t \mathbf{v} = 0$ for every $\alpha \in S$. Then $h(\mathbf{y} + t\mathbf{v}) = h(\mathbf{y})$ for every scalar t , which is incompatible with $h(\mathbf{y})$ being strictly convex.

Strict convexity guarantees uniqueness, not existence, of a minimum point. Coerciveness ensures existence. The objective function $f(\mathbf{x})$ is coercive if $f(\mathbf{x})$ tends to ∞ whenever any component of \mathbf{x} tends to 0 or ∞ . Under the reparameterization $x_i = e^{y_i}$, this is equivalent to $h(\mathbf{y}) = f(\mathbf{x})$ tending to ∞ as $\|\mathbf{y}\|_2$ tends to ∞ . A necessary and sufficient condition for this to occur is that $\max_{\alpha \in S} \alpha^t \mathbf{v} > 0$ for every $\mathbf{v} \neq \mathbf{0}$. For a proof, suppose the contrary condition holds for some $\mathbf{v} \neq \mathbf{0}$. Then it is clear that $h(t\mathbf{v})$ remains bounded above by $h(\mathbf{0})$ as the scalar t tends to ∞ . Conversely, if the stated condition is true, then the function $q(\mathbf{y}) = \max_{\alpha \in S} \alpha^t \mathbf{y}$ is continuous and achieves its minimum of $d > 0$ on the sphere $\{\mathbf{y} \in \mathbb{R}^n : \|\mathbf{y}\|_2 = 1\}$. It follows that $q(\mathbf{y}) \geq d\|\mathbf{y}\|_2$ and that

$$h(\mathbf{y}) \geq \max_{\alpha \in S} \{c_\alpha e^{\alpha^t \mathbf{y}}\} \geq \left(\min_{\alpha \in S} c_\alpha \right) e^{d \|\mathbf{y}\|_2}.$$

This lower bound shows that $h(\mathbf{y})$ is coercive.

The coerciveness condition is hard to apply in practice. An equivalent condition is that the origin $\mathbf{0}$ belongs to the interior of the convex hull of the set $\{\alpha\}_{\alpha \in S}$. It is straightforward to show that the negations of these two conditions are logically equivalent. Thus, suppose $q(\mathbf{v}) = \max_{\alpha \in S} \alpha^t \mathbf{v} = 0$ for some vector $\mathbf{v} \neq \mathbf{0}$. Every convex combination $\sum_{\alpha} p_\alpha \alpha$ then satisfies $(\sum_{\alpha} p_\alpha \alpha)^t \mathbf{v} = 0$. If the origin is in the interior of the convex hull, then $\varepsilon \mathbf{v}$ is also for every sufficiently small $\varepsilon > 0$. But this leads to the contradiction $\varepsilon \mathbf{v}^t \mathbf{v} = \varepsilon \|\mathbf{v}\|_2^2 \leq 0$. Conversely, suppose $\mathbf{0}$ is not in the interior of the convex hull. According to the separating hyperplane theorem for convex sets, there exists a unit vector \mathbf{v} with $\mathbf{v}^t \alpha = 0 = \mathbf{v}^t \mathbf{0}$ for every $\alpha \in S$. In other words, $q(\mathbf{v}) = 0$. The convex hull criterion is easier to check, but it is not constructive. In simple cases such as the objective function $f_1(\mathbf{x})$ where the power vectors are $\alpha = (-3, 0)^t$, $\alpha = (-1, -2)^t$, and $\alpha = (1, 1)^t$, it is visually obvious that the origin is in the interior of their convex hull.

One can also check the criterion $q(\mathbf{v}) > 0$ for all $\mathbf{v} \neq \mathbf{0}$ by solving a related geometric programming problem. This problem consists in minimizing the scalar t subject to the inequality constraints $\alpha^t \mathbf{y} = t$ for all $\alpha \in S$ and the nonlinear equality constraint $\|\mathbf{y}\|_2^2 = 1$. If $t_{\min} = 0$, then the original criterion fails.

In some cases, the objective function $f(\mathbf{x})$ does not attain its minimum on the open domain $\mathbb{R}_{>0}^n = \{\mathbf{x} : x_i > 0, 1 \leq i \leq n\}$. This condition is equivalent to the corresponding function $h(\mathbf{y})$ being unbounded below on \mathbb{R}^n . According to Gordon's theorem [2,10], this can happen if and only if $\mathbf{0}$ is not in the convex hull of the set $\{\alpha\}_{\alpha \in S}$. Alternatively, both conditions are equivalent to the existence of a vector \mathbf{v} with $\alpha^t \mathbf{v} < 0$ for all $\alpha \in S$. For the objective function $f_3(\mathbf{x})$, the power vectors are $\alpha = (-1, -2)^t$ and $\alpha = (1, 1)^t$. The origin $(0, 0)^t$ does not lie on the line segment between them, and the vector $(-3/2, 1)^t$ forms a strictly oblique angle with each. As predicted, $f_3(\mathbf{x})$ does not attain its infimum on $\mathbb{R}_{>0}^n$.

The theoretical development in reference [10] demonstrates that the MM algorithm converges at a linear rate to the unique minimum point of the objective function $f(\mathbf{x})$ when $f(\mathbf{x})$ is coercive and its convex reparameterization $h(\mathbf{y})$ is strictly convex. The theory does not cover other cases, and it would be interesting to investigate them. The general convergence theory of MM algorithms [10] states that five properties of the objective function $f(\mathbf{x})$ and MM algorithmic map $\mathbf{x} \mapsto M(\mathbf{x})$ guarantee convergence to a stationary point of $f(\mathbf{x})$: (a) $f(\mathbf{x})$ is coercive on its open domain; (b) $f(\mathbf{x})$ has only isolated stationary points; (c) $M(\mathbf{x})$ is continuous; (d) \mathbf{x}^* is a fixed point of $M(\mathbf{x})$ if and only if \mathbf{x}^* is a stationary point of $f(\mathbf{x})$; and (e) $f[M(\mathbf{x}^*)] = f(\mathbf{x}^*)$, with equality if and only if \mathbf{x}^* is a fixed point of $M(\mathbf{x})$. For a general signomial program, items (a) and (b) are the hardest to check. Our examples provide some clues.

The standard convergence results for the quadratic penalty method are covered in the references [1,10,13,17]. To summarize the principal finding, suppose that the objective function $f(\mathbf{x})$ and the constraint functions $r_i(\mathbf{x})$ and $s_i(\mathbf{x})$ are continuous and that $f(\mathbf{x})$ is coercive on $\mathbb{R}_{>0}^n$. If \mathbf{x}_λ minimizes the penalized objective function

$$f_\lambda(\mathbf{x}) = f(\mathbf{x}) + \lambda \sum_i r_i(\mathbf{x})^2 + \lambda \sum_j s_j(\mathbf{x})_+^2,$$

and \mathbf{x}_∞ is a cluster point of \mathbf{x}_λ as λ tends to ∞ , then \mathbf{x}_∞ minimizes $f(\mathbf{x})$ subject to the constraints. In this regard observe that the coerciveness assumption on $f(\mathbf{x})$ implies that the solution set $\{\mathbf{x}_\lambda\}_\lambda$ is bounded and possesses at least one cluster point. Of course, if the solution set consists of a single point, then \mathbf{x}_λ tends to that point.

8 Discussion

The current paper presents novel algorithms for both geometric and signomial programming. Although our examples are low dimensional, the previous experience of Sha et al. [18] offers convincing evidence that the MM algorithm works well for high-dimensional quadratic programming with nonnegativity constraints. The ideas pursued here – the MM principle, separation of variables, quasi-Newton acceleration, and penalized optimization – are surprisingly potent in large-scale optimization. The MM algorithm deals with the objective function directly and reduces multivariate minimization to a sequence of one-dimensional minimizations. The MM updates are simple to code and enjoy the crucial descent property. Treating constrained signomial programming by the penalty method extends the MM algorithm even further. Quadratic programming with linear equality and inequality constraints is the most important special case of constrained signomial programming. Our new MM algorithm for constrained quadratic programming deserves consideration in high-dimensional problems. Even though MM algorithms can be notoriously slow to converge, quasi-Newton acceleration can dramatically improve matters. Acceleration involves no matrix inversion, only matrix times vector multiplication. In our limited experiments with large-scale problems [22,23], MM algorithms with quasi-Newton acceleration can achieve comparable or better performance than limited-memory BFGS algorithms. Finally, it is worth keeping in mind that parameter separated algorithms are ideal candidates for parallel processing.

Because geometric programs are convex after reparameterization, it is relatively easy to pose and check sufficient conditions for global convergence of the MM algorithm. In contrast it is far more difficult to analyze the behavior of the MM algorithm for signomial programs. Theoretical progress will probably be piecemeal and require problem-specific information. A major difficulty is understanding the asymptotic nature of the objective function as parameters approach 0 or ∞ . Even in the absence of theoretical guarantees, the descent property of the MM algorithm makes it an attractive solution technique and a diagnostic tool for finding counterexamples. Some of our test problems expose the behavior of the MM algorithm in non-standard situations. We welcome the help of the optimization community in unraveling the mysteries of the MM algorithm in signomial programming.

Acknowledgments

Research was supported by United States Public Health Service grants GM53275, MH59490, and R01HG006139.

References

1. Bertsekas, DP. *Nonlinear Programming*. Athena Scientific; 1999.
2. Borwein, JM.; Lewis, AS. *Convex Analysis and Nonlinear Optimization: Theory and Examples*. New York: Springer-Verlag; 2000.
3. Boyd S, Kim S-J, Vandenberghe L, Hassibi A. A tutorial on geometric programming. *Optimization and Engineering*. 2007; 8:67–127.

4. Boyd, S.; Vandenberghe, L. *Convex Optimization*. Cambridge: Cambridge University Press; 2004.
5. Ecker JG. Geometric programming: methods, computations and applications. *SIAM Review*. 1980; 22:338–362.
6. Feigin PD, Passy U. The geometric programming dual to the extinction probability problem in simple branching processes. *Annals Prob*. 1981; 9:498–503.
7. del Mar Hershenson M, Boyd SP, Lee TH. Optimal design of a CMOS op-amp via geometric programming. *IEEE Trans Computer-Aided Design*. 2001; 20:1–21.
8. Hoffman, K. *Analysis in Euclidean Space*. Englewood Cliffs, NJ: Prentice-Hall; 1975.
9. Hunter DR, Lange K. A tutorial on MM algorithms. *Amer Statistician*. 2004; 58:30–37.
10. Lange, K. *Optimization*. New York: Springer-Verlag; 2004.
11. Lange K, Hunter DR, Yang I. Optimization transfer using surrogate objective functions (with discussion). *J Comput Graphical Stat*. 2000; 9:1–59.
12. Mazumdar M, Jefferson TR. Maximum likelihood estimates for multinomial probabilities via geometric programming. *Biometrika*. 1983; 70:257–261.
13. Nocedal, J.; Wright, SJ. *Numerical Optimization*. Springer; 1999.
14. Passy U, Wilde DJ. A geometric programming algorithm for solving chemical equilibrium problems. *SIAM J Appl Math*. 1968; 16:363–373.
15. Peressini, AL.; Sullivan, FE.; Uhl, JJ, Jr. *The Mathematics of Nonlinear Programming*. New York: Springer-Verlag; 1988.
16. Peterson EL. Geometric programming. *SIAM Review*. 1976; 18:338–362.
17. Ruszczyński, A. *Optimization*. Princeton University Press; 2006.
18. Sha, F.; Saul, LK.; Lee, DD. Multiplicative updates for nonnegative quadratic programming in support vector machines. In: Becker, S.; Thrun, S.; Obermayer, K., editors. *Advances in Neural Information Processing Systems*. Vol. 15. Cambridge, MA: MIT Press; p. 1065-1073.
19. Steele, JM. *The Cauchy-Schwarz Master Class: An Introduction to the Art of Inequalities*. Cambridge: Cambridge University Press and the Mathematical Association of America; 2004.
20. Wang Y, Zhang K, Shen P. A new type of condensation curvilinear path algorithm for unconstrained generalized geometric programming. *Math. Comput. Modelling*. 2002; 35:1209–1219.
21. Zhou H, Alexander D, Lange KL. A quasi-Newton acceleration method for high-dimensional optimization algorithms. *Statistics and Computing*. 2011; 21:261–173. [PubMed: 21359052]
22. Zhou H, Lange KL. MM algorithms for some discrete multivariate distributions. *Journal of Computational and Graphical Statistics*. 2010; 19(3):645–665. [PubMed: 20877446]
23. Zhou H, Lange KL. A fast procedure for calculating importance weights in bootstrap sampling. *Computational Statistics and Data Analysis*. 2011; 55:26–33. [PubMed: 21076658]
24. Zhou H, Lange KL. Path following in the exact penalty method of convex programming. *arXiv: 1201.3593*. 2011
25. Zhou H, Lange KL, Suchard MA. Graphical processing units and high-dimensional optimization. *Statistical Science*. 2010; 25(3):311–324. [PubMed: 21847315]

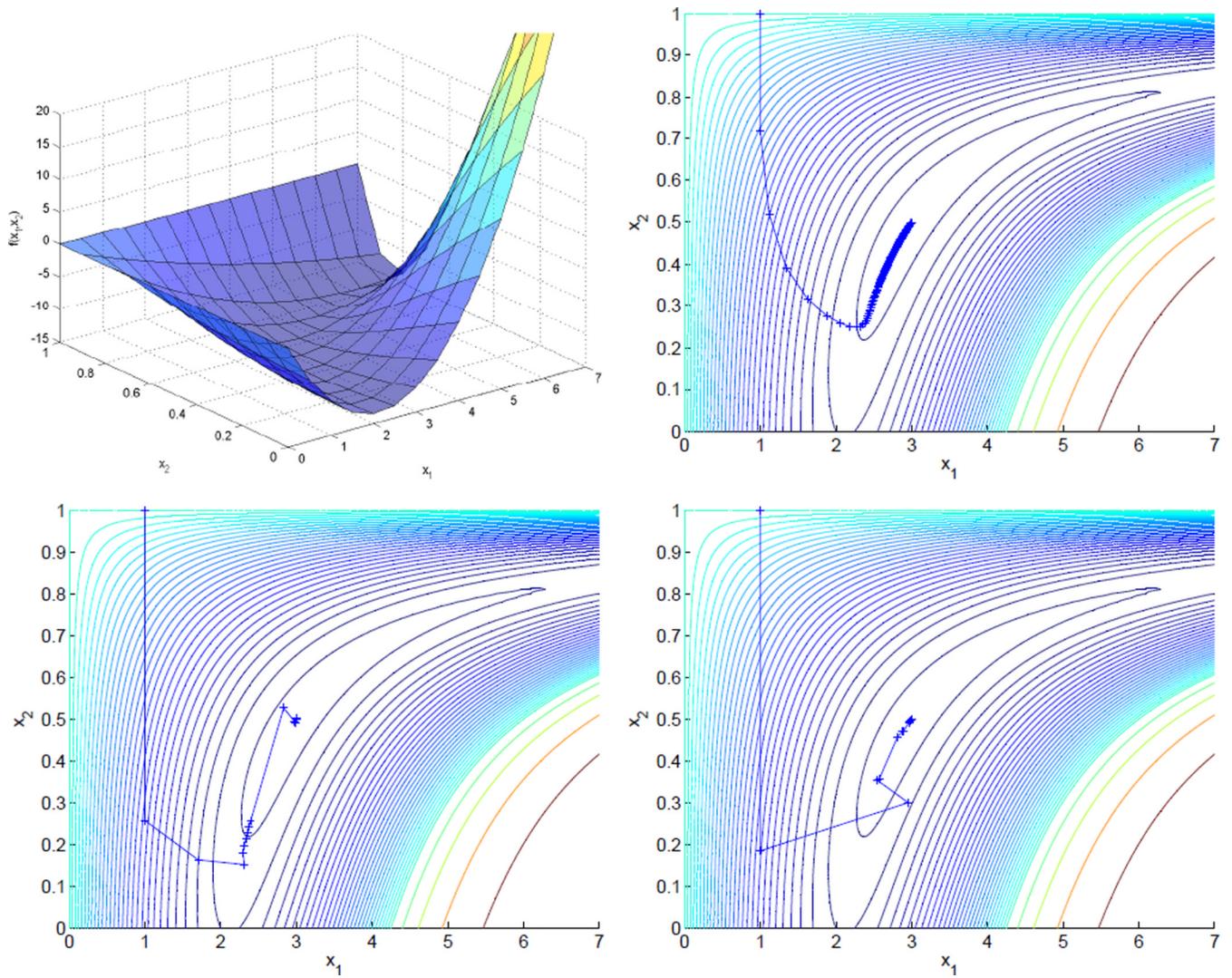


Fig. 1. Upper left: The test function $f_6(x)$. Upper right: 558 MM iterates. Lower left: 30 accelerated MM iterates ($q = 1$ secant conditions). Lower right: 12 accelerated MM iterates ($q = 2$ secant conditions).

Table 1

Numerical examples of unconstrained signomial programming. Test functions $f_4(\mathbf{x}), f_6(\mathbf{x}), f_7(\mathbf{x}), f_8(\mathbf{x})$ and $f_9(\mathbf{x})$ are taken from [20]. P: posynomial; S: signomial; G: general function.

Fun	Type	Initial Point x_0	Min Point	Min Value	Itrs (10^{-6})
f_1	P	(1,2)	(1.4310,1.4310)	3.4128	38
f_2	P	(1,2)	(0.6300,1.2599)	2.0000	2
f_3	P	(1,1)	diverges	0.0000	
f_4	S	(0.1,0.2,0.3,0.4)	(0.1596,0.3191,0.1954,0.2606)	0.0000	3
f_5	G	(1,1,1)	(0.4082,0.4082,0.4082)	0.2973	2
		(1,2,3)	diverges	$-\infty$	
f_6	S	(1,1)	(2.9978,0.4994)	-14.2031	558
f_7	S	(1, ..., 10)	$0.0255x_0$	0.0000	18
f_8	P	(1, ..., 7)	diverges	0.0000	
f_9	P	(1,2,3,4)	(0.3969,0.0000,0.0000,1.5874)	2.0000	7

Table 2

Iterates from the quadratic penalty method for the test function $f_{10}(x)$. The convergence criterion for the inner loops is 10^{-9} .

$\log_2 \lambda$	Iters	x_λ
0	8	(0.9503,1.6464)
1	6	(0.8580,1.5164)
2	5	(0.8138,1.4461)
3	23	(0.7853,1.4067)
4	32	(0.7264,1.3702)
5	31	(0.6967,1.3518)
6	30	(0.6817,1.3426)
7	29	(0.6742,1.3380)
8	28	(0.6704,1.3356)
9	26	(0.6686,1.3345)
10	25	(0.6676,1.3339)
11	23	(0.6671,1.3336)
12	22	(0.6669,1.3335)
13	21	(0.6668,1.3334)
14	19	(0.6667,1.3334)
15	18	(0.6667,1.3334)
16	16	(0.6667,1.3333)
17	15	(0.6667,1.3333)

Table 3

Iterates from the quadratic penalty method for the test function $f_{11}(\mathbf{x})$. The convergence criterion for the inner loops is 10^{-16} .

$\log_2 \lambda$	Iters ($q = 0$)	Iters ($q = 1$)	x_λ
0	18	5	(3.0000,1.8000)
1	2	2	(2.8571,1.7143)
2	56	6	(2.6667,1.6667)
3	97	5	(2.5455,1.6364)
4	167	5	(2.4762,1.6190)
5	312	5	(2.4390,1.6098)
6	541	6	(2.4198,1.6049)
7	955	5	(2.4099,1.6025)
8	1674	4	(2.4050,1.6012)
9	2924	3	(2.4025,1.6006)
10	4839	3	(2.4013,1.6003)
11	7959	4	(2.4006,1.6002)
12	12220	4	(2.4003,1.6001)
13	17674	4	(2.4002,1.6000)
14	21739	3	(2.4001,1.6000)
15	20736	3	(2.4000,1.6000)
16	8073	3	(2.4000,1.6000)
17	111	3	(2.4000,1.6000)
18	6	4	(2.4000,1.6000)
19	5	2	(2.4000,1.6000)
20	3	2	(2.4000,1.6000)
21	2	2	(2.4000,1.6000)