AN ALTERNATIVE METHOD FOR CONSTRAINED OPTIMIZATION

PABLO PEDREGAL

Abstract. We introduce an alternative approach for constrained mathematical programming problems. It rests on two main aspects: an efficient way to compute optimal solutions for unconstrained problems, and multipliers regarded as variables for a certain map. Contrary to typical dual strategies, optimal vectors of multipliers are sought as fixed points for that map. Two distinctive features are worth highlighting: its simplicity and flexibility for the implementation, and its convergence properties.

1. Introduction

We are concerned here with the general, standard mathematical program

\begin{equation}
\text{Minimize in } x \in \mathbb{R}^N: \quad f(x) \text{ subject to } h(x) = 0, g(x) \leq 0,
\end{equation}

for a smooth, real function $f : \mathbb{R}^N \to \mathbb{R}$, and smooth, vector-valued mappings $h : \mathbb{R}^N \to \mathbb{R}^n$, $g : \mathbb{R}^N \to \mathbb{R}^m$. Karush-Kuhn-Tucker (KKT) optimality conditions are among the basic techniques taught, and learnt, in optimization courses (there are hundreds of textbooks on the subject, see for instance [3]). They involved, in addition to $x$ itself, multipliers $z \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, for all of the constraints in the problem to be respected. Under appropriate constraint qualifications, that are not part of our discussion here, local solutions of (1.1) are to be found among the triplets $(x, y, z) \in \mathbb{R}^N \times \mathbb{R}^m \times \mathbb{R}^n$ complying with

\begin{equation}
\nabla f(x) + z \cdot \nabla h(x) + y \cdot \nabla g(x) = 0,
\end{equation}

\begin{equation}
h(x) = 0, \quad y \cdot g(x) = 0,
\end{equation}

\begin{equation}
y \geq 0, \quad g(x) \leq 0.
\end{equation}

These optimality conditions furnish fundamental insight and information into the solutions of (1.1). Under standard sets of constraint qualifications, solutions of optimality conditions furnish (local) solutions of (1.1). They are the guiding principle to design numerical algorithms to approximate those solutions. They are also the starting point of duality theory so fundamental to the understanding of mathematical programming. As it is well-known, the basic idea of duality is to set up a new mathematical program, intimately connected to (1.1), and, in particular, designed with the main ingredients of that (primal) problem, but in which multipliers $(y, z)$ play a central role in the form of dual variables. There is an intimate relationship between optimal solutions $x$ of the primal, and optimal solutions $(\bar{y}, \bar{z})$ of the dual.

Our point of view here is a bit different, and though it also deals with multipliers $(y, z)$, we seek them, in association with the primal variable $x$, not in the form of the optimal solution of another (dual) mathematical program, but rather as a fixed point of a suitable map. What is more important, the structure of that map is such that, under mild assumptions, a typical iterative procedure consisting in iterating the action of such map, converges to the optimal triplet $(x, \bar{y}, \bar{z})$. It is remarkable that the map is so simple to define, and so easy to implement in practice, if we can rely on an efficient procedure for unconstrained optimization.

E.T.S. Ingenieros Industriales. Universidad de Castilla La Mancha. Campus de Ciudad Real (Spain). Research supported by MTM2013-47053-P of the Mineco (Spain). e-mail: pablo.pedregal@uclm.es.
The strategy of using free, unconstrained programs to approximate the optimal solutions of general constrained optimization problems like \((1.1)\) is quite natural, and appealing. This is the source of many fundamental algorithms utilized today. It will also be a main inspiration for us here, as indicated before.

The method we want to examine is designed to deal just with inequality constraints so that there is no map \(h\) in \((1.1)\)

\[
\begin{align*}
(1.2) \quad & \text{Minimize in } x \in \mathbb{R}^N: \quad f(x) \quad \text{subject to} \quad g(x) \leq 0, \\
\end{align*}
\]

for a smooth, real function \(f: \mathbb{R}^N \to \mathbb{R}\), and a smooth, vector-valued mapping \(g: \mathbb{R}^N \to \mathbb{R}^m\). This is not, in principle, a significant limitation because the equality constraint \(h(x) = 0\) can be, equivalently, translated either into \(-h(x) \leq 0\) together with \(-h(x) \leq 0\), or else into \(g_0(x) \leq 0\) for \(g_0(x) = (1/2)|h(x)|^2\). We will therefore stick to problem \((1.2)\).

A central role in our method is played by the master function

\[
L(x, y) = f(x) + \sum_{k=1}^m e^{y(k)} g_k(x), \quad y = \left( y^{(k)} \right)_{k=1,2,...,m}, \quad g = (g_k)_{k=1,2,...,m}.
\]

It is definitely reminiscent of the typical Lagrangean for dual theory, though at the same time it is a bit different. We will understand soon the main reasons that support such choice. Notice that this master function is quite different from \(f(x) + \exp(y \cdot g(x))\). We will comment on this later.

Suppose that \(L(x, y)\) is strictly convex, and coercive in \(x\) for every choice \(y \in \mathbb{R}^m\) of vectors with (strictly) positive coordinates. Our basic map

\[
G(y): \mathbb{R}^m_+ \mapsto \mathbb{R}^m_+
\]

is the result of the composition of two steps:

1. For given \(y \in \mathbb{R}^m_+\), find (approximate) the (global) solution of the unconstrained problem

   Minimize in \(x \in \mathbb{R}^N:\quad L(x,y).

   The passage from \(y\) to \(x \equiv x(y)\) is, therefore, a well-defined and smooth operation.

2. Put

   \[
   G = (G_k)_{k=1,2,...,m}, \quad G_k(y) = y^{(k)} e^{y^{(k)} g_k(x(y))},
   \]

   when \(y \in \mathbb{R}^m_+\), and extend it by continuity for \(y \in \mathbb{R}^m_+\).

This extension by continuity deserves some comments. On the one hand, note that if some component \(k\) of \(y\) vanishes, \(y^{(k)} = 0\), then trivially \(G_k(y) = 0\) regardless of the value of \(g_k(y)\). This is not convenient, since \(y^{(k)} = 0\) must be somehow related to the constraint \(g_k(y) \leq 0\). On the other, notice that, after all, the constraint \(g_k(x) \leq 0\) is equivalent to \(y^{(k)} g_k(x) \leq 0\) for every positive \(y^{(k)}\), but there is definitely a discontinuity if we set \(y^{(k)} = 0\), for then the constraint drops out. In other words, the optimization problem

\[
\begin{align*}
\text{Minimize in } x \in \mathbb{R}^N: \quad f(x) \quad \text{subject to} \quad y^{(k)} g_k(x) \leq 0 \quad \text{for all } k,
\end{align*}
\]

for a fixed vector \(y\) with (strictly) positive components \(y^{(k)}\) is equivalent to \((1.2)\). However, if some of the components of \(y\) vanish, then the corresponding constraint drops out, and so there is clearly a lack of continuity.

A somewhat surprising fact that places this map \(G\) into perspective is the following.

**Proposition 1.1.** Suppose a certain vector \(y \in \mathbb{R}^m_+\) is a fixed point for \(G\). Then \(x = x(y)\) is a (local) solution of \((1.2)\).
Proof. If \( \mathbf{y} \) is a true fixed point for \( \mathbf{G} \), then we should have
\[
\mathbf{y}^{(k)} = \mathbf{y}^{(k)} e^{g_k(\mathbf{x})}.
\]
This equation amount to two possibilities: either \( y^{(k)} = 0 \), or else \( 1 = e^{g_k(\mathbf{x})} \), i.e., \( y^{(k)} g_k(\mathbf{x}) = 0 \). At any rate, \( y^{(k)} g_k(\mathbf{x}) = 0 \) for all \( k \).

Suppose that we could find a vector \( \mathbf{x} \), not far from \( \mathbf{x} \), such that \( f(\mathbf{x}) \leq f(\mathbf{x}) \) and \( g(\mathbf{x}) \leq 0 \). Because \( \mathbf{y} \geq 0 \), we would have \( y^{(k)} g_k(\mathbf{x}) \leq 0 = y^{(k)} g_k(\mathbf{x}) \) for all \( k \). Hence, it is clear that
\[
L(\mathbf{x}, \mathbf{y}) < L(\mathbf{x}, \mathbf{y}),
\]
but this contradicts the very nature of \( \mathbf{x} \) as a local minimum for \( L(\cdot, \mathbf{y}) \).

It is interesting to notice that one can even allow for a fixed point for \( \mathbf{G} \) “at infinity” with the same proof. This situation is particularly important for us because our way to deal with equality constraints will lead us to this case.

**Proposition 1.2.** Suppose that \( y_j \in \mathbb{R}^m_+ \) is such that \( \|G(y_j) - y_j\| \to 0 \), and \( x_j = x(y_j) \to x \in \mathbb{R}^N \). Then \( x \) is a local solution of \([1, 2]\).

Proof. Just as in the proof of the Proposition 1.1, one would conclude that \( y_j^{(k)} g_k(x_j) \to 0 \) as \( j \to \infty \) for all \( k \).

Suppose that we could find a vector \( \mathbf{x} \), not far from \( \mathbf{x} \), such that \( f(\mathbf{x}) < f(\mathbf{x}) \) and \( g(\mathbf{x}) \leq 0 \). Again, we would have \( y_j^{(k)} g_k(\mathbf{x}) \leq 0 \) for all \( j \), and \( k \). Put
\[
-2\epsilon = f(\mathbf{x}) - f(\mathbf{x}) > 0.
\]
By continuity of \( f \), for all \( j \) sufficiently large (depending on \( \epsilon \)),
\[
-\epsilon = f(\mathbf{x}) - f(\mathbf{x}) > 0.
\]
For \( j \) even larger if necessary, we will have
\[
e^{y_j^{(k)} g_k(\mathbf{x})} \leq \frac{\epsilon}{2m} + e^{y_j^{(k)} g_k(\mathbf{x})},
\]
for all \( k \). Then, for those large \( j \)’s,
\[
f(\mathbf{x}) + \sum_k e^{y_j^{(k)} g_k(\mathbf{x})} = -\epsilon + f(\mathbf{x}) + \sum_k e^{y_j^{(k)} g_k(\mathbf{x})}
\leq -\frac{\epsilon}{2} + f(\mathbf{x}) + \sum_k e^{y_j^{(k)} g_k(\mathbf{x})}
\leq f(\mathbf{x}) + \sum_k e^{y_j^{(k)} g_k(\mathbf{x})},
\]
but the resulting inequality contradicts the nature of \( \mathbf{x} \) as a local minimum for \( L(\cdot, \mathbf{y}_j) \).

Note that, if for some component \( k \), \( y_j^{(k)} \to \infty \), optimality conditions would break down as there would be no multiplier for that component.

The numerical procedure that arises from these propositions is amazingly simple to implement, but relies in a fundamental way on being capable of efficiently approximating the map \( x(y) \). This amounts to unconstrained optimization. It reads:

1. **Initialization.** Take \( y_0 \in \mathbb{R}^m_+ \), \( y_0 > 0 \), and \( x_0 \in \mathbb{R}^N \) in an arbitrary way, or appropriately located in a certain valley. For instance, \( y_0 = 1 \), \( x_0 = 0 \).
2. **Iterative step until convergence.** Suppose we have computed \( y_j \), and \( x_j \).
(a) Solve for the unconstrained optimization problem

\[
\text{Minimize in } z \in \mathbb{R}^N: \quad L(z, y_j) = f(z) + \sum_{k=1}^{m} e^{y_j^{(k)} g_k(z)}
\]

starting from the initial guess \( x_j \). Let \( x_{j+1} \) be such (local) minimizer.

(b) If \( y_j \cdot g(x_{j+1}) \) vanishes (i.e. is reasonably small), stop, take \( y_j \) as the multiplier of the problem, and \( x_{j+1} \) as the solution of the constrained problem.

(c) If \( y_j \cdot g(x_{j+1}) \) does not vanish, update

\[
y_j^{(k)} + 1 = e^{y_j^{(k)} g_k(x_{j+1})} y_j^{(k)}
\]

for all \( k = 1, 2, \ldots, m \).

The intuition after this algorithm is pretty clear. Each value \( y \in \mathbb{R}^m \) establishes an exponential barrier for the constraints, in such a way that if the minimizer \( x(y) \) turns out to be non-feasible, then the barrier should be intensified. This is what the update rule \((1.4)\) does in that case. If, on the other hand, the constraint is met, then the barrier should be relaxed so as letting the objective function \( f \) to seek “more freely without restriction” its minimum. This is again accomplished by the update rule.

This is a good place to stress how the form of \( L(x, y) \) cannot be \( f(x) + \exp(y \cdot g(x)) \), because for this other choice, the update rule for the auxiliary variable \( y \)

\[
y_{j+1} = e^{y_j \cdot g(x_{j+1})} y_j
\]

would be the same for all components, and this is too rigid to work well: each component \( k \) should adapt to its corresponding constraint separately from the others.

Beyond the convergence theorems that we will prove, Propositions \((1.1)\) and \((1.2)\) are clear and powerful statements. In practice, without any further concern about assumptions, one can use the above algorithm. If the variable \( x \) does converge (even if \( y \) does not), the limit vector \( \bar{x} \) has to be a (local) solution of \((1.2)\). Indeed, the algorithm is quite flexible to the point that the set-valued map \( y \mapsto X(y) \), where \( X(y) \) stands for the full set of local minima of \( L(\cdot, y) \), admits selections to approximate all of the (isolated) local minima of \((1.1)\).

Our main task here focuses on showing that, under appropriate standard hypotheses, this algorithm always converges to minima of the underlying constrained problem \((1.2)\).

**Theorem 1.3.** Suppose the cost function \( f : \mathbb{R}^N \rightarrow \mathbb{R} \), and the components of the constraint map \( g : \mathbb{R}^N \rightarrow \mathbb{R}^m \) comply with:

1. they all are smooth, and convex;
2. the corresponding \( L(x, y) \) is coercive in \( x \) for every fixed \( y \) with positive components;
3. for all \( \bar{y} > 0 \) with components sufficiently large, the minimizer \( \bar{x} \) of \( L(x, \bar{y}) \) with respect to \( x \) is such that \( g(\bar{x}) \leq 0 \).

Then the above algorithm always converges to a (global) minimizer for \((1.2)\).

The last hypothesis in the statement of this result may be hard to be examined on specific examples. Fortunately, when there are optimal solutions for the underlying mathematical programming problem, that condition need not be checked explicitly.

**Corollary 1.4.** Suppose the cost function \( f : \mathbb{R}^N \rightarrow \mathbb{R} \), and the components of the constraint map \( g : \mathbb{R}^N \rightarrow \mathbb{R}^m \) comply with:

1. they all are smooth, and convex, and \( f \) is bounded from below;
2. the corresponding \( L(x, y) \) is coercive in \( x \) for every fixed \( y \) with positive components;
3. the program \((1.2)\) determined by \( f \) and \( g \) admits optimal solutions.

Then the above algorithm always converges to a (global) minimizer for \((1.2)\).
By removing the convexity conditions, we are typically left with a local convergence theorem. Different local solutions are reached by different initializations in the algorithm, as remarked before the statement. The practical implication of our analysis, as suggested earlier, is that optimal solutions of mathematical programs will be captured by this algorithm whenever they exist.

The main goal of the paper is, in addition to introducing the algorithm itself, to prove this convergence theorem. We will proceed through several steps of increasing generality, devoting some attention to the situation where some equality constraints are to be enforced, and a final, short section to linear programming.

A fundamental next step is to test the practical performance of the algorithm for specific problems. Another main topic for the near future is to extend the algorithm to continuous (infinite-dimensional) optimization problems under constraints of different kinds.

The only place we have found where some reference is made to this point of view is [1]. The authors report that it was T. S. Motzkin who, in 1952, suggested the use of exponentials for satisfying a system of linear inequalities, though, the authors of [1] report, it was not applied to develop a usable algorithm. One of our favorites sources for numerical optimization is [2].

2. A PERSPECTIVE FOR CONSTRAINED PROBLEMS BASED ON UNCONSTRAINED MINIMIZATION

We will start with the simple basic problem

(2.1) \[ \text{Minimize in } x \in \mathbb{R}^N : f(x) \text{ subject to } g(x) \leq 0, \]

where both \( f \) and \( g \) are smooth functions. We therefore have a single inequality constraint.

We would like to design an iterative procedure to approximate solutions for (2.1) in an efficient, practical, accurate way. We introduce our initial demands in the form:

1. The main iterative step is to be an unconstrained minimization problem, and as such its corresponding objective function must be defined in all of space (exterior point methods).
2. More specifically, we would like to design a real function \( H \) so that the main iterative step of our procedure be applied to the augmented cost function \( L(y, x) = f(x) + H(yg(x)) \) for the \( x \)-variable. We hope to take advantage of the joint dependence upon \( y \) and \( x \) inside the argument for \( H \). Notice that letting \( y \) out of \( H \) may not mean a real change as we would be back to (2.1) with a \( g \) which would be the composition \( H(g(x)) \).
3. The passage from one iterative step to the next is performed through an update step for the variable (multiplier) \( y \).
4. Convergence of the scheme should lead to a solution of (2.1).

Notice that

(2.2) \[ \nabla_x L(y, x) = \nabla f(x) + H'(yg(x))y \nabla g(x), \]

and that optimality conditions for (2.1) read

(2.3) \[ \nabla f(x) + y \nabla g(x) = 0, \quad yg(x) = 0, \quad y \geq 0, \quad g(x) \leq 0. \]

Thus each main iterative step enforces the main equation in (2.3), the one involving gradients and derivatives. But we would like to design the function \( H(t) \) to ensure that as a result of the iterative procedure, the other conditions in (2.3) are also met.

The following features seem to be very convenient:

1. Variable \( x \) will always be a solution of

\[ \nabla f(x) + H'(yg(x))y \nabla g(x) = 0. \]
Variable $y$ will always be non-negative (in practice strictly positive but possibly very small). Comparison of (2.2) with (2.3) leads to the identification $y \mapsto H'(yg(x))y$, and so we would like $H' > 0$.

The multiplier $H'(yg(x))y$ can only vanish if $y$ does. The update rule for the variable $y$ should be $y \mapsto H'(yg(x))y$. If at some step we hit the true value of the multiplier $y$, then simultaneously $yg(x) = 0$, and so we would also like to have $H'(0) = 1$.

If $g(x) > 0$, then the update rule above for $y$ must yield a higher value for $y$ so as to force $g(x)$ into the next iterative step the feasible inequality $g \leq 0$. Hence, $H'' > 0$, or $H$, convex (for positive values). In addition, $H''(t) \to +\infty$ when $t \to +\infty$.

If $g(x)$ turns out to be (strictly) negative, then we would like $y$ to become smaller so as to let the minimization of $f$ proceed with a lighter interference from the inequality constraint. This again leads to $G$ convex (for negative values).

The optimality condition $yg(x) = 0$ becomes $H'(yg(x))yg(x) = 0$. Thus the function $H'(t)\to$ can only vanish if $t = 0$. In particular, $H' > 0$.

All of these reasonable conditions impose the requirements

\[ H' > 0, \quad H'(0) = 1, \quad H'' > 0, \quad H''(t) \to \infty \text{ if } t \to \infty. \]

Possibly, the most familiar choice if $H(t) = e^t$, and this is the one we will select.

The iterative procedure is then as follows.

1. Initialization. Take $y_0 > 0$, and $x_0 \in \mathbb{R}^N$ in an arbitrary way. For instance, $y_0 = 1$, $x_0 = 0$.

2. Iterative step until convergence. Suppose we have $y_j, x_j$.
   (a) Solve for the unconstrained optimization problem
   \[
   \text{Minimize in } z \in \mathbb{R}^N : \quad f(z) + e^{y_jg(z)}
   \]
   starting from the intial guess $x_j$. Let $x_{j+1}$ be such (local) minimizer.
   (b) If $y_jg(x_{j+1})$ vanishes, stop: take $y_j$ as the multiplier of the problem, and $x_{j+1}$ as the solution of the constrained problem.
   (c) If $y_jg(x_{j+1})$ does not vanish, update $y_{j+1} = e^{y_jg(x_{j+1})}y_j$.

3. A convergence theorem

We would like to provide a solid foundation for our approximation procedure for mathematical programs by proving convergence theorems as the following. Recall that the master function is

\[ L(x, y) = f(x) + e^{yg(x)}. \]

Though the treatment of this one-dimensional situation is elementary, it will be the basic building block upon which prove the general, multidimensional case.

**Theorem 3.1.** Suppose the cost function $f : \mathbb{R}^N \to \mathbb{R}$, and the constraint function $g : \mathbb{R}^N \to \mathbb{R}$ satisfy the following requirements:

1. they are smooth, and convex;
2. the corresponding $L(x, y)$ is coercive in $x$ for every fixed positive $y$;
3. for some $\tilde{y} > 0$ large and fixed, the minimizer $\tilde{x}$ of $L(x, \tilde{y})$ with respect to $x$ is such that $g(\tilde{x}) \leq 0$.

Then the above algorithm always converges to a (global) minimizer for (2.1).
Proof. By a standard perturbation argument depending on a small parameter \( \epsilon > 0 \), we can assume, without loss of generality, that the convexity condition imposed on \( f \) is strict, and that the hessian \( \nabla^2 f(x) \) is a symmetric, positive definite matrix for all \( x \). It suffices to add a term like \((\epsilon/2)|x|^2\) to \( f(x)\). Under this strengthened hypothesis, the master function \( L \), regarded as a function of \( x \), is a coercive, strictly convex function, and so the unique minimizer \( x \equiv x(y) \) is determined implicitly through the (unique) solution of the non-linear system

\[
\nabla f(x) + e^{yg(x)}y\nabla g(x) = 0, \tag{3.1}
\]

and so it is smooth by the implicit function theorem. Notice that the gradient of \( f \), with respect to \( x \) (the hessian of \( L(\cdot, y) \)) cannot be singular precisely because \( L(x, y) \) is strictly convex with respect to \( x \). The conclusion of the statement will be a direct consequence of two lemmata whose proofs rely on suitable manipulations of \( (3.1) \). All functions involved are smooth.

Lemma 3.2. For every positive \( y \),

\[
[1 + yg(x(y))] \frac{d}{dy} g(x(y)) \leq 0. \tag{3.2}
\]

Proof. Since \( x(y) \) is determined as the result of a (local) minimization process, we also have, in addition to \( (3.1) \),

\[
\nabla^2 f(x) + e^{yg(x)} y^2 \nabla g(x) \nabla g(x) + e^{yg(x)} y \nabla^2 g(x) \geq 0
\]
as symmetric matrices. This condition is nothing but the positivity of the hessian (with respect to \( x \) as has already been indicated above. In particular, since all functions are smooth, \( x'(y) \) is well-defined, and

\[
x' \cdot \nabla^2 f(x)x' + e^{yg(x)} y^2 |\nabla g(x)x'|^2 + e^{yg(x)} y x' \cdot \nabla^2 g(x)x' \geq 0. \tag{3.3}
\]

On the other hand, from

\[
\nabla f(x(y)) + e^{yg(x(y))}y\nabla g(x(y)) = 0,
\]
differentiating with respect to \( y \), we arrive at

\[
\nabla^2 f(x)x' + e^{yg(x)}[g(x) + y\nabla g(x)x']y\nabla g(x) + e^{yg(x)}[\nabla g(x) + y\nabla^2 g(x)x'] = 0.
\]

Multiplying by \( x' \), and comparing to \( (3.2) \), we see that

\[
e^{yg(x)}[yg(x) + 1]\nabla g(x)x' \leq 0.
\]

This is exactly the statement in the lemma. \( \square \)

Let us now focus on the function \( G(y) : \mathbb{R}^+ \to \mathbb{R}^+ \) given by

\[
G(y) = e^{yg(x(y))}y
\]

where \( x(y) \) is again determined by \( (3.1) \), and \( y > 0 \).

Lemma 3.3. The function \( G \) is smooth, and for every positive \( y \),

\[
G'(y) \frac{d}{dy} g(x(y)) \leq 0.
\]

Proof. It is clear that \( (3.1) \) can be written as

\[
\nabla f(x(y)) + G(y)\nabla g(x(y)) = 0.
\]

By differentiating with respect to \( y \), we will have

\[
\nabla^2 f(x)x'(y) + G'(y)\nabla g(x) + G(y)\nabla^2 g(x)x'(y) = 0, \tag{3.3}
\]

and

\[
x'(y) \cdot \nabla^2 f(x)x'(y) + G'(y)x'(y) \cdot \nabla g(x) + G(y)x'(y) \cdot \nabla^2 g(x)x'(y) = 0.
\]
Therefore
\[
G'(y) \frac{d}{dy} g(x(y)) = -x'(y) \cdot [\nabla^2 f(x) + G(y)\nabla^2 g(x)]' \leq 0
\]
due to the convexity assumed on \( f \), and \( g \).

If we further put, for simplicity, \( \overline{f}(y) = g(x(y)) \), we have the three properties
\[
(3.4) \quad (1 + y\overline{f}(y))\overline{f}'(y) \leq 0, \quad G'(y)\overline{f}'(y) \leq 0, \quad G'(y)(1 + y\overline{f}(y)) \geq 0.
\]
Notice that the third one is a consequence of the other two, which are the conclusion of the two lemmata above. From these properties, we would like to highlight the following consequences:

1. If \( \overline{f}(\overline{y}) < 0 \), then \( \overline{f}(y) < 0 \) for all \( y \geq \overline{y} \). This is a direct consequence of the first inequality in (3.4). Indeed, suppose there is some \( y > \overline{y} \) with \( \overline{f}(y) \geq 0 \), and put
   \[ y_0 = \inf \{ y > \overline{y} : \overline{f}(y) \geq 0 \}. \]
   It is then clear that \( \overline{f}(y_0) = 0 \), and \( \overline{f}(y_0) \geq 0 \). But then, again that first inequality in (3.4) would imply \( \overline{f}(y_0) \leq -1/y_0 \), a contradiction. The argument is standard. As a consequence if \( G(\overline{y}) < \overline{y} \), then \( G(y) < y \) for all \( y \geq \overline{y} \). By continuity we can incorporate the equality as well: if \( G(\overline{y}) \leq \overline{y} \), then \( G(y) \leq y \) for all \( y \geq \overline{y} \).

2. Over the set \( \overline{f} \geq 0 \), i.e. over the set where \( G \) is greater than or equal to the identity, \( G \) is non-decreasing (third inequality in (3.4)), and \( \overline{f} \) is non-increasing (first inequality in (3.4)).

We can have two main scenarios. Notice that the last hypothesis in the statement implies that \( G(\overline{y}) \leq \overline{y} \), and by the first conclusion above, \( G(y) \leq y \) for all \( y \geq \overline{y} \).

1. It could happen that \( G(y) < y \), for all \( y > 0 \) (Figure 1, left picture). In this case, \( G \) has a unique fixed point in \( y = 0 \), and it is globally stable. This corresponds to the situation where the global minimizer of \( f \) complies with \( g \leq 0 \), and the restriction is inactive.

2. Assume there is some \( y_0 > 0 \), so that \( G(y_0) > y_0 \) (Figure 1, right picture). By the properties just written, \( G \) will exactly have one fixed point in the interval \( [y_0, \overline{y}] \), and the iteration process will always, regardless of the starting point, converge to such a fixed point of \( G \). In this case \( \overline{f} \) will vanish at such fixed point, and the restriction is active.
4. THE CASE OF MULTIPLE CONSTRAINTS

Once the nature of the algorithm we would like to propose has been clarified with a single inequality constraint, we want to examine the same strategy for several such conditions. It will suffice to focus on just two such constraints to be able to figure out the situation with many more such inequality constraints.

Consider the optimization problem

\[(4.1) \quad \text{Minimize in } x \in \mathbb{R}^N : \quad f(x) \quad \text{subject to} \quad g_1(x) \leq 0, \quad g_2(x) \leq 0.\]

After the situation examined above, we focus on the unconstrained problem

Minimize in \( x \in \mathbb{R}^N : \quad f(x) + e^{y_1 g_1(x)} + e^{y_2 g_2(x)}, \)

and regard the (local) solution \( x \equiv x(y_1, y_2) \) for \( y_i > 0 \) as an implicit mapping depending on \( y = (y_1, y_2) \). We assume that \( f \) as well as \( g_i \) are such that there is no difficulty in finding \( x(y_1, y_2) \), and that this dependence is continuous, even smooth. After Lemma \[3.2\] we would like to use the fundamental information

\[
\nabla f(x) + e^{y_1 g_1(x)} y_1 \nabla g_1(x) + e^{y_2 g_2(x)} y_2 \nabla g_2(x) = 0,
\]

together with the local convexity condition

\[
\nabla^2 f(x) + e^{y_1 g_1(x)} [y_1^2 \nabla^2 g_1(x) \nabla g_1(x) + y_1 \nabla^2 g_1(x)] + e^{y_2 g_2(x)} [y_2^2 \nabla^2 g_2(x) \nabla g_2(x) + y_2 \nabla^2 g_2(x)] \geq 0,
\]

in the sense of symmetric matrices. One would have then to fix a unit direction \( n = (n_1, n_2) \), and try to find relevant information much in the same way as we have done with the single inequality situation. We believe, however, that it is a much more transparent strategy to freeze alternatively each one of the two multiplier \( y_i \), \( i = 1, 2 \), and apply the single inequality constraint with respect to the complementary constraint.

Namely, let \( y_1 \geq 0 \) be fixed, and consider the mathematical program

Minimize in \( x \in \mathbb{R}^N : \quad f(x) + e^{y_1 g_1(x)} \quad \text{subject to} \quad g_2(x) \leq 0.\)

Assume that the hypotheses of Theorem \[3.1\] permit us to conclude that there is \( x^{(1)} \equiv x(y_1) \), and \( y_2^{(1)} \equiv y_2(y_1) \) such that

\[
\nabla f(x^{(1)}) + e^{y_1 g_1(x^{(1)})} y_1 \nabla g_1(x^{(1)}) + y_2^{(1)} \nabla g_2(x^{(1)}) = 0,
\]

\[
y_2^{(1)} \geq 0, \quad g_2(x^{(1)}) \leq 0, \quad y_2^{(1)} g_2(x^{(1)}) = 0.
\]

Likewise, we would also have \( x^{(2)} \equiv x(y_2) \), and \( y_1^{(2)} \equiv y_1(y_2) \) such that

\[
\nabla f(x^{(2)}) + y_1^{(2)} \nabla g_1(x^{(2)}) + e^{y_2 g_2(x^{(2)})} y_2 \nabla g_2(x^{(2)}) = 0,
\]

\[
y_1^{(2)} \geq 0, \quad g_1(x^{(2)}) \leq 0, \quad y_1^{(2)} g_1(x^{(2)}) = 0.
\]

This solution would correspond to the problem

Minimize in \( x \in \mathbb{R}^N : \quad f(x) + e^{y_2 g_2(x)} \quad \text{subject to} \quad g_1(x) \leq 0, \)

again through Theorem \[3.1\] assuming that the appropriate hypotheses hold. The whole argument then revolves around ensuring that the graphs of the two functions \( y_2(y_1) \), and \( y_1(y_2) \) meet at some pair \( \bar{y} = (\bar{y}_1, \bar{y}_2) \), for in this case we would have a vector

\[
x = x(\bar{y}_1, \bar{y}_2) = x^{(1)}(\bar{y}_1) = x^{(2)}(\bar{y}_2).
\]
such that
\begin{equation}
\nabla f(x) + \bar{y}_1 \nabla g_1(x) + \bar{y}_2 \nabla g_2(x) = 0,
\end{equation}
\[
\bar{y} \geq 0, \quad g(x) \leq 0, \quad \bar{y} \cdot g(x) = 0, \quad \bar{y} = (\bar{y}_1, \bar{y}_2),
\]
a solution of the mathematical program (4.1). The iterative procedure can be set up in a similar way.

(1) Initialization. Take \( y_0 > 0 \) and \( x_0 \in \mathbb{R}^N \) in an arbitrary way. For instance, \( y_0 = 1, x_0 = 0 \).

(2) Iterative step until convergence. Suppose we have \( y_j, x_j \).
(a) Solve for the unconstrained optimization problem
\[
\text{Minimize in } z \in \mathbb{R}^N : \quad f(z) + \sum_{k=1}^{2} e^{y_j^{(k)}} g_k(z)
\]
starting from the initial guess \( x_j \). Let \( x_{j+1} \) be such (local) minimizer.
(b) If \( y_j \cdot g(x_{j+1}) \) vanishes, stop and take \( y_j \) as the multiplier of the problem, and \( x_{j+1} \) as the solution of the constrained problem.
(c) If \( y_j \cdot g(x_{j+1}) \) does not vanish, update
\[
y_{j+1}^{(k)} = e^{y_j^{(k)}} g_k(x_{j+1}) y_j^{(k)}
\]
for \( k = 1, 2 \).

Keep in mind the two functions \( y_1(y_2) \), and \( y_2(y_1) \), as defined in the discussion above: \( y_1(y_2) \) corresponds to the multiplier for the one-dimensional situation (Section 3) with objective functional \( f(x) + e^{y_2 g_2(x)} \), and constraint \( g_1(x) \leq 0 \). Similarly for the other one.

**Theorem 4.1.** Suppose the cost function \( f : \mathbb{R}^N \to \mathbb{R} \), and the two components of the constraint map \( g : \mathbb{R}^N \to \mathbb{R}^2 \) comply with:

1. they all are smooth, and convex;
2. the corresponding \( L(x, y) \) is coercive in \( x \) for every \( y \) with (strictly) positive components;
3. for every \( \bar{y} > 0 \) with sufficiently large components, the minimizer \( \bar{x} \) of \( L(x, \bar{y}) \) with respect to \( x \) is such that \( g(x) < 0 \) (both components).

Then the above algorithm always converges to a solution of (4.1) which is a (global) minimizer for (4.1).

**Proof.** Again, we may assume through a standard perturbation argument, and without loss of generality, that \( f \) is strictly convex. In this way, the master function \( L(x, y) \) is coercive and strictly convex in \( x \), for every \( y > 0 \), and the mapping taking each \( y \) into the unique (global) minimizer \( x \equiv x(y) \) of \( L(x, y) \) with respect to \( x \) is well-defined, and smooth.

Let us consider the smooth mapping \( G : \mathbb{R}^2_+ \to \mathbb{R}^2_+ \) carrying \( y \) into
\[
y \circ e^{y \cdot g(x(y))} = \left( y_{1}^{(k)} e^{y^{(k)} g_k(x(y))} \right)_{k=1,2}.
\]

Our statement is equivalent to saying that there is a unique fixed point for \( G \) which is globally stable in \( \mathbb{R}^2_+ \).

Let us focus on the vectors \( \bar{y} = (\bar{y}_1, \bar{y}_2) \) in the statement. Take one such vector \( \bar{y} = (\bar{y}_1, \bar{y}_2) \) with \( \bar{y}_1 > y_2(0), \bar{y}_2 > y_1(0) \), where the functions \( y_1 \) and \( y_2 \) have been introduced above. Recall that \( y_i(0) \) is the multiplier associated with the program
\[
\text{Minimize in } x \in \mathbb{R}^N : \quad f(x) \quad \text{subject to} \quad g_i(x) \leq 0,
\]
for $i = 1, 2$. We then claim that $\tilde{y}_1 > y_1(\tilde{y}_2)$, and, similarly, $\tilde{y}_2 > y_2(\tilde{y}_1)$. In fact, by definition of $y_1(\tilde{y}_2)$, we should have that $g_1(y_1(\tilde{y}_2), \tilde{y}_2) = 0$, while $g_1(\tilde{y}_1, \tilde{y}_2) < 0$ by hypothesis. By the discussion with the single-inequality constraint in the proof of Theorem 3.1 (right after the proof of Lemma 3.3), this implies the claim. Likewise, for the other case. This conclusion, together with the previous choice of $\tilde{y}$, immediately implies that the graphs of the two functions intersect (at least) in a point $(\tilde{y}_1, \tilde{y}_2) \in \mathbb{R}^2_+$ (the closure of $\mathbb{R}^2_+$). As indicated earlier in the discussion before the statement of the theorem, this intersection point generates a solution of (4.2). We can even take $\tilde{y}$, with larger components if necessary, so that $\tilde{y}_1 > y_1(y_2)$ for all $y_2$ in the interval $[0, \tilde{y}_2]$, and $\tilde{y}_2 > y_2(y_1)$ for all $y_1 \in [0, \tilde{y}_1]$.

Let us now deal with the convergence issue. It is based on the global convergence for the one dimensional situation, and from this point of view is not difficult to show. We proceed in two easy steps. Refer to Figure 2.

(1) Because of our final choice of the vector $\tilde{y}$, it is clear that the whole sequence of iterates is uniformly bounded because each update rule

$$y_k \mapsto \bar{z}_k = \bar{y}_k e^{\bar{y}_k g_k(x)}, \quad k = 1, 2,$$

(4.3)
tends to the graph of the corresponding function $y_2(y_1)$, or $y_1(y_2)$, for both components, respectively, according to the discussion prior to the statement of the theorem. In fact, this argument implies already the convergence. Iterates stay in the box $[0, \tilde{y}_1] \times [0, \tilde{y}_2]$.

(2) Assume then that we have a certain iterate $(\bar{y}, \bar{x})$ where $\bar{x} = x(\bar{y})$. Again by the remarks made before the statement of the theorem, the difference $\bar{y}_1 - z_1$ in (4.3) corresponds to the one-dimensional process when the second variable $y_2$ is frozen at the value $\bar{y}_2$. Likewise for the difference $\bar{y}_2 - z_2$ when $y_1 = \bar{y}_1$. By the uniform boundedness claimed in the previous point, and by continuity, those differences then converge to zero.

\[ \square \]
We finally come to the general situation in which we become interested in the general mathematical program

\[(4.4) \text{Minimize in } x \in \mathbb{R}^N : f(x) \text{ subject to } g(x) \leq 0,\]

where \( g = (g_k) : \mathbb{R}^N \rightarrow \mathbb{R}^m \). The fundamental map around which revolves our algorithm takes a vector of multiplier \( y = (y^{(k)}) \in \mathbb{R}^m, \ y > 0, \) into a (local) solution \( x(y) \) of our basic unconstrained problem

\[(4.5) \text{Minimize in } x \in \mathbb{R}^N : f(x) + \sum_{k=1}^{m} e^{y^{(k)}g_k(x)} .\]

The objective function

\[L(x, y) = f(x) + \sum_{k=1}^{m} e^{y^{(k)}g_k(x)}\]

of this unconstrained problem is the master function of the problem.

KKT optimality conditions read

\[(4.6) \nabla f(x) + y \nabla g(x) = 0, \ y \cdot g(x) = 0, \ y \geq 0, \ g(x) \leq 0,\]

and we would like to show that our main algorithm always converges to some local solution of \[(4.4)\].

The algorithm has already been described:

1. Initialization. Take \( y_0 \in \mathbb{R}^m, \ y_0 > 0, \) and \( x_0 \in \mathbb{R}^N \) in an arbitrary way. For instance, \( y_0 = 1, \ x_0 = 0. \)
2. Iterative step until convergence. Suppose we have computed \( y_j \) and \( x_j \).
   a. Solve the unconstrained optimization problem
      \[\text{Minimize in } z \in \mathbb{R}^N : L(z, y_j) = f(z) + \sum_{k=1}^{m} e^{y^{(k)}g_k(z)}\]
      starting from the initial guess \( x_j \). Let \( x_{j+1} \) be such (local) minimizer.
   b. If \( y_j \cdot g(x_{j+1}) \) vanishes, stop, take \( y_j \) as the multiplier of the problem, and \( x_{j+1} \) as the solution of the constrained problem.
   c. If \( y_j \cdot g(x_{j+1}) \) does not vanish, update
      \[y^{(k)}_{j+1} = e^{y^{(k)}g_k(x_{j+1})} y^{(k)}_j\]
      for all \( k = 1, 2, \ldots, m \).

The proof of Theorem 1.3 follows exactly the same strategy as with the two-component case.

5. Equality constraints

As has been sufficiently emphasized before, the nature of our approach to mathematical programming prevent us to deal directly with problems where equality constraints are to be enforced. When this is the situation, equality constraints have to be replaced by an equivalent set of inequality constraints. Two typical possibilities of doing this has already been pointed out in the Introduction. Though the structure of specific problems may recommend different ways of going through this transformation, the equivalence

\[ h(x) = 0 \iff g_0(x) \leq 0, \ g_0(x) = \frac{1}{2} |h(x)|^2, \]

will probably be a favorite possibility in most of the examples. In this way all of the equality constraints become a single, additional inequality constraint.
To understand better the difficulties about the application of our convergence Theorem 1.3 to such a situation, let us investigate a typical mathematical program with only equality constraints like

Minimize in $x \in \mathbb{R}^N$: $f(x)$ subject to $h(x) = 0$,

which can be recast in the form

Minimize in $x \in \mathbb{R}^N$: $f(x)$ subject to $g(x) \leq 0$,

for

$$g(x) = \frac{1}{2} |h(x)|^2.$$  

There are two serious difficulties with this last problem.

1. The first one is about optimality conditions. The condition involving gradients is

$$\nabla f(x) + \mu h(x) \nabla h(x) = 0.$$  

But since feasible vectors must comply with $h(x) = 0$, that condition is void: there is no multiplier for the problem written in this form. This situation takes us to Proposition 1.2.

2. A local version of Theorem 1.3 (without the convexity conditions) could not be applied as the last hypothesis would be impossible to hold precisely because $g(x) \leq 0$ means $h(x) = 0$, and so, that assumption would require that the solution $\tilde{x}$ of the problem

$$\min_{x \in \mathbb{R}^N} f(x) + e^y g(x)$$  

would always be feasible $h(\tilde{x}) = 0$ for $y$ large. It is not reasonable to expect this. It will probably be true that $g(\tilde{x})$ is small for $y$ large, but not zero.

We need therefore to relax that assumption in Theorem 1.3. Though there might be various ways to do it, we focus on the one that we believe is more appealing from a practical viewpoint. This is Corollary 1.4. We restate here.

**Corollary 5.1.** Suppose the cost function $f: \mathbb{R}^N \to \mathbb{R}$, and the components of the constraint map $g: \mathbb{R}^N \to \mathbb{R}^m$ comply with:

1. they all are smooth, and convex, and $f$ is bounded from below;
2. the corresponding $L(x, y)$ is coercive in $x$ for every fixed $y$ with positive components;
3. the program (1.2) determined by $f$ and $g$ admits optimal solutions.

Then the above algorithm always converges to a (global) minimizer for (1.2).

**Proof.** We proceed in two steps. Let $\mathbf{x}$ be a solution of the program (1.2), and $M$ a lower bound for $f$.

Step 1. We first claim that

$$\liminf_{\min_k y_k \to \infty} g(\mathbf{x}(y)) = 0.$$  

Suppose, seeking a contradiction, that there is a fixed $\delta > 0$ so that

(5.1) $\|g(\mathbf{x}(y))\| \geq \delta$

whenever $\min_k y_k$ is large. For all $y$, we should have

$$M + \sum_k e^{y(k)} g_k(\mathbf{x}(y)) \leq f(\mathbf{x}(y)) + \sum_k e^{y(k)} g_k(\mathbf{x}(y))$$

$$= \min_x \left[ f(x) + \sum_k e^{y(k)} g_k(x) \right]$$

$$\leq f(\mathbf{x}) + M,$$
because \( g_k(x) \leq 0 \). Recall that \( m \) is the number of constraints.

But under (5.1), the left-hand side converges to infinity if \( \min_k y_k \to \infty \), a contradiction.

**Step 2.** The coercivity of the master function

\[
    f(x) + \sum_k \epsilon^{y(k)} g_k(x)
\]

is uniform over the non-feasible set \( g(x) \geq 0 \), because the exponentials are increasing functions of \( y > 0 \) for each fixed, non-feasible \( x \). This fact implies that the family \( \{x(y)\} \) is uniformly bounded, and so every accumulation point will be a solution of our mathematical program according to Proposition 1.2. Notice that \( \|G(y) - y\| \to 0 \) is exactly the conclusion of Step 1. \( \square \)

It is interesting to remark that the conclusion of Step 1 of the previous proof together with our main convergence Theorem 1.3 imply that for each \( \epsilon > 0 \), one can always find optimal solutions of the parametrized family of problems

Minimize in \( x \in \mathbb{R}^N : f(x) \) subject to \( g(x) \leq \epsilon 1 \),

where \( 1 = (1,1,\ldots,1) \).

## 6. Linear Programming

We would also like to explicitly discussed the situation of a linear program, for some hypotheses hold due to the nature of such problems. Consider

\[(6.1) \quad \text{Minimize in } x \in \mathbb{R}^N : \quad u \cdot x \quad \text{subject to } \quad Ax \leq b,\]

where vectors \( u \in \mathbb{R}^N \), \( b \in \mathbb{R}^m \) are constant, and \( A \in \mathbb{R}^{m \times N} \) is also a constant matrix. Our master, auxiliary function is

\[
    L(x,y) = u \cdot x + \sum_{k=1}^m \epsilon^{y(k)} (A_k \cdot x - b_k),
\]

where \( A_k \) are the rows of \( A \), and \( b = (b_k) \). The following lemma does not require further comments.

**Lemma 6.1.** Assume \( L \) is a coercive function of \( x \) for each fixed \( y > 0 \). Then (6.1) admits optimal solutions.

First-order optimality conditions read

\[(6.2) \quad u + \sum_{k=1}^m \epsilon^{y(k)} (A_k \cdot x - b_k) y^{(k)} A_k = 0.\]

This (non-linear) system determines \( x \) as an implicit map of \( y \). Let us examine the update rule for \( y \). Suppose \( y = (y^{(k)}) \) is known. Solve for, or approximate, \( x \) as a solution of (6.2) by minimizing \( L(x,y) \) with respect to \( x \). The update rule is

\[ y^{(k)} \mapsto \epsilon^{y^{(k)}(A_k \cdot x - b_k)} y^{(k)} \]

for all \( k \). This algorithm converges to the (one) solution of the linear program.

**Corollary 6.2.** Suppose, as in the previous lemma, that \( L(x,y) \) is coercive in \( x \) for every \( y > 0 \). Then our algorithm always (for arbitrary initialization) converges to one optimal solution of (6.1).
REFERENCES

[1] Fiacco, A. V., McCormick, G. P., Nonlinear Programming. Sequential Unconstrained Minimization Techniques, SIAM Classics in Appl. Math., 4, Philadelphia.

[2] Nocedal, J., Wright, S. J. Numerical optimization, Springer Series in Operations Research, Springer-Verlag, New York, 1999.

[3] Hiriart-Urruty, J. B., Lemarchal, C., Convex analysis and minimization algorithms. I. Fundamentals, Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], 305. Springer-Verlag, Berlin, 1993.