

Operations Research Techniques and Algorithms (620-361)

Dr Yao-ban Chan

y.chan@ms.unimelb.edu.au

Telephone: 8344 9073

Office: Room 198, Richard Berry Building

Christina Burt

c.burt@ms.unimelb.edu.au

Telephone: 8344 1797

Office: 139 Barry St

Friday 2nd May, 2008

In practice, we would solve the unconstrained problem by numerical methods, and so it is impossible to take the exact limit as $k \rightarrow \infty$. What we would normally do is to progressively set the penalty parameter higher and higher, and when the algorithm stops, take the nearest feasible point as a solution.

By slowly increasing the penalty parameter, we can also derive appropriate starting points for each phase of unconstrained algorithm.

We next present a proof of the penalty function theorem in the case with only equality constraints. Suppose x' solves the non-linear program, so for all x such that $h(x) = 0$,

$$f(x') \leq f(x).$$

From the statement of the theorem, x^k minimises the penalty function $P_{\alpha_k}(x)$. Therefore we know that

$$P_{\alpha_k}(x^k) \leq P_{\alpha_k}(x').$$

Expanding this out gives

$$f(x^k) + \frac{\alpha_k}{2} \sum_{j=1}^q [h_j(x^k)]^2 \leq f(x') + \frac{\alpha_k}{2} \sum_{j=1}^q [h_j(x')]^2 = f(x')$$

because x' is feasible for NLP. Rearranging gives

$$\sum_{j=1}^q [h_j(x^k)]^2 \leq \frac{2}{\alpha_k} (f(x') - f(x^k)).$$

If x^* is the limit point of x^k , then we can take the limit on both sides as $k \rightarrow \infty$ to get

$$\sum_{j=1}^q [h_j(x^*)]^2 \leq \lim_{k \rightarrow \infty} \frac{2}{\alpha_k} (f(x') - f(x^k)) = 0.$$

The last limit follows because $f(x') - f(x^k)$ is finite and $\alpha_k \rightarrow \infty$ as $k \rightarrow \infty$. This tells us that $h_j(x^*) = 0$ for all j , and therefore x^* is feasible.

To prove that x^* is optimal, we note that from before,

$$f(x^k) + \frac{\alpha_k}{2} \sum_{j=1}^q [h_j(x^k)]^2 \leq f(x').$$

Taking the limit as $k \rightarrow \infty$ gives us

$$f(x^*) \leq f(x^*) + \lim_{k \rightarrow \infty} \frac{\alpha_k}{2} \sum_{j=1}^q [h_j(x^k)]^2 \leq f(x').$$

Since x' is an optimal solution for the NLP, and x^* has an equal or lower objective function value, we conclude that x^* is also an optimal solution for the NLP.

l_2 -penalty methods are just one of the many types of penalty methods that we can apply to constrained problems, and like most other methods it has advantages and disadvantages.

One of the more obvious disadvantages is that the l_2 penalty function is not C^2 , so we cannot apply Newton's method to solve the unconstrained penalty problem.

We can overcome this problem by using a smoother function.

The *log barrier* penalty method does this by taking the penalty function

$$P_\alpha(x) = f(x) - \frac{1}{\alpha} \sum_i \log(-g_i(x)) + \frac{\alpha}{2} \sum_j [h_j(x)]^2.$$

For the logarithm to be defined, this requires that $-g_i(x) > 0$, which means $g_i(x) < 0$, i.e. the point is strictly feasible.

Now as $g_i(x) \rightarrow 0$, $-\frac{1}{\alpha} \log(-g_i(x)) \rightarrow \infty$, so the boundary of the feasible region acts as a barrier preventing the solution from going past it — hence the name!

The only problem with this is that this does not allow $g_i(x)$ to be 0, which we want, because as $g_i(x) \rightarrow 0$, $-\frac{1}{\alpha} \log(-g_i(x)) \rightarrow \infty$. However, as $\alpha \rightarrow \infty$, this term becomes smaller and smaller, so with “ $\alpha = \infty$ ”, we can have $g_i(x) = 0$.

But since all points leading up to the limit must have been feasible because of the barrier, the limit point must be feasible too!

It turns out that the log-barrier method has a similar convergence result to the l_2 penalty method theorem.

Theorem. Let f , g and h be C^1 . Suppose that x^* is a KKT point of the NLP which is a local minimum. Then there exists a sequence of points x^k such that x^k is a local minimum of $P_{\alpha_k}(x)$, and $x^k \rightarrow x^*$ as $k \rightarrow \infty$ (and therefore $\alpha_k \rightarrow \infty$).

As before, we can find estimates of the KKT multipliers by taking the limit of certain quantities in the log-barrier problem.

$$\begin{aligned}\nabla P_\alpha(x) &= \nabla f(x) - \sum_i \frac{1}{\alpha} \nabla(\log(-g_i(x))) + \sum_j \frac{\alpha}{2} \nabla[h_j(x)]^2 \\ &= \nabla f(x) - \sum_i \frac{1}{\alpha g_i(x)} \nabla g_i(x) + \sum_j \alpha h_j(x) \nabla h_j(x) \\ &= 0\end{aligned}$$

when the penalty function is minimised.

By comparing with KKTa as before, we can see that if we set

$$-\frac{1}{\alpha g_i(x^k)} \rightarrow \lambda_i^*$$

and

$$\alpha h_j(x^k) \rightarrow \eta_j^*$$

as $k \rightarrow \infty$, then λ^* and η^* give the KKT multipliers.

Again, we can look a bit closer to uncover some of KKTb. Because $\alpha > 0$ and $g_i(x^k) < 0$ (remember all the x^k are strictly feasible), we can say that $\lambda_i^* \geq 0$.

Furthermore, if $g_i(x^*) < 0$, the α term in the denominator of λ_i^* will go to ∞ and ensure that $\lambda_i^* = 0$. On the other hand, if $\lambda_i^* > 0$, then this means that $\alpha g_i(x^k)$ cannot tend to ∞ , and the only way this can happen is if $g_i(x^k) \rightarrow 0$ — in other words, $g_i(x^*) = 0$. As before, this is encapsulated in $\lambda_i^* g_i(x^*) = 0$.

Example. Apply the log-barrier method to the l_2 penalty method example:

$$\begin{array}{ll} \min & x_1^2 + x_2^2 + x_1 - x_2 \\ \text{s.t.} & x_1 \geq 1 \\ & x_2 \geq 0. \end{array}$$



