Theoretical Economics Letters, 2011, 1, 8-14
doi:10.4236/tel.2011.11003 Published Online May 2011 (http://www.SciRP.org/journal/tel)

Decrease of the Penalty Parameter in Differentiable Penalty Function Methods

Roohollah Aliakbari Shandiz[1], Emran Tohidi[2]
[1]*Faculty of Mathematical Sciences, Sharif University of Technology, Tehran, Iran*
[2]*Department of Applied Mathematics, Ferdowsi University of Mashhad, Mashhad, Iran*
E-mail: aliakbari_r@mehr.sharif.ir, emran.tohidi@stu-mail.um.ac.ir
Received March 30, 2011; *revised April* 28, 2011; *accepted May* 4, 2011

Abstract

We propose a simple modification to the differentiable penalty methods for solving nonlinear programming problems. This modification decreases the penalty parameter and the ill-conditioning of the penalty method and leads to a faster convergence to the optimal solution. We extend the modification to the augmented Lagrangian method and report some numerical results on several nonlinear programming test problems, showing the effectiveness of the proposed approach.

Keywords: Nonlinear Programming, Penalty Method, Penalty Parameter, Differentiable Penalty Methods

1. Introduction

Solving nonlinear programming (*NLP*) problems via a penalty method was first introduced by Courant [1] in 1943. Fiacco and McCormick [2] developed barrier methods for solving *NLP* problem. Murray [3] show that the Hessian matrix of penalty method is ill-con- ditioned. Since then, many approaches for reducing the ill-conditioning of penalty method were proposed. To avoid too increasing of the penalty parameter, Zangwill [4] introduced exact nondifferentiable penalty functions and Fletcher [5] introduced continuously differentiable exact penalty functions. Another exact penalty methods have been studied in [6-13] and others. In addition, Mongeau [14] decreased the penalty parameter in exact penalty methods for solving linear programming problems. Here, Using general ideas of Mongeau, we propose an approach to reduce the penalty parameter in the differentiable penalty method for solving *NLP* problems.

2. The Basic Idea

Consider the following programming problem:

$$(NLP)\qquad \begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_j(x)\le 0, \quad j=1,\cdots,m, \\ & x\in X, \end{array}$$

where f and the g_j are twice continuously differentiable functions.

Let $\mu>0$, a common penalty function for (*NLP*) is:

$$(H1)\qquad H_1^\mu(x)=f(x)+\mu P(x),$$

where, $P(x)=\sum_{j=1}^m g_j^+(x)^2$ and $g_j^+(x)=\max(0,g_j(x))$.

A penalty problem for (*NLP*) is defined as follows:

$$(PEN1)\qquad \begin{array}{ll} \min & H_1^\mu(x)=f(x)+\mu P(x) \\ \text{s.t.} & x\in X. \end{array}$$

Gradient and hessian of H_1^μ can be calculated as follows:

$$\nabla H_1^\mu(x)=\nabla f(x)+\mu\nabla P(x)=\nabla f(x)+2\mu\sum_{j=1}^m g_j^+(x)\nabla g_j(x),$$

$$\nabla^2 H_1^\mu(x)=\nabla^2 f(x)+\mu\nabla^2 P(x)=\nabla^2 f(x)+2\mu\sum_{j=1}^m g_j^+(x)\nabla^2 g_j(x)+2\mu\sum_{j=1}^m \nabla g_j^+(x)\nabla^T g_j^+(x).$$

Let $U(x)=2\sum_{j=1}^m g_j^+(x)\nabla^2 g_j(x)$ and $V(x)=2\sum_{j=1}^m \nabla g_j^+(x)\nabla^T g_j^+(x)$. Thus,

$$\nabla^2 H_1^{\mu}(x) = \nabla^2 f(x) + \mu U(x) + \mu V(x). \qquad (2.1)$$

Note that due to the continuity of second derivatives, Hessian matrices $\nabla^2 f$, $\nabla^2 P$ and $\nabla^2 g_j$ are symmetric.

The condition number of a square matrix $\boldsymbol{A}$ is given by $K(\boldsymbol{A}) = \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|$. If $K(\boldsymbol{A})$ is large, then $\boldsymbol{A}$ is said to be ill-conditioned. For a symmetric matrix $\boldsymbol{A}$, it can be shown that

$$K(\boldsymbol{A}) = \lambda_{\max}/\lambda_{\min},$$

where, $\lambda_{\max}$ and $\lambda_{\min}$ are the largest and smallest eigenvalues of matrix $\boldsymbol{A}$, respectively.

If we assume that there are r active constraints at x^*, the optimal solution of (*NLP*), and The gradients of these constraints are linearly independent, Then V has rank equal to r and thus has r nonzero eigenvalues. (2.1) implies that when $\mu \to \infty$ at least r eigenvalues of $\nabla^2 H_1^{\mu}$ tend to infinity. It has been shown in [15] that exactly r eigenvalues tend to infinity and $n-r$ other eigenvalues tend to finite limits, which implies the ill-conditioning of the Hessian of penalty method.

To avoid the ill-conditioning, instead of usual penalty function we consider the following function:

$$(H2) \qquad H_2^{\mu}(x) = f(x)/\mu + P(x), \qquad \mu > 0.$$

Its corresponding penalty problem for (*NLP*) is:

$$(PEN\,2) \qquad \begin{array}{ll} \min & H_2^{\mu}(x) = f(x)/\mu + P(x) \\ \text{s.t.} & x \in X. \end{array}$$

It is easy to see that problems (*PEN*1) and (*PEN*2) are equivalent. Because $H_1^{\mu}(x) = \mu H_2^{\mu}(x)$.

Gradient and Hessian of H_2^{μ} is

$$\nabla H_2^{\mu}(x) = \nabla f(x)/\mu + \nabla P(x),$$

$$\begin{aligned}\nabla^2 H_2^{\mu}(x) &= \nabla^2 f(x)/\mu + \nabla^2 P(x) \\ &= \nabla^2 f(x)/\mu + U(x) + V(x).\end{aligned}$$

If $\nabla^2 P(x^*)$ is of full rank (for example, if P is a strictly convex function), then all eigenvalues of $\nabla^2 P(x^*)$ are nonzero. Thus, for a large enough μ all eigenvalues of $\nabla^2 H_2^{\mu}(x^*)$ are also nonzero. Therefore unlike H_1^{μ}, H_2^{μ} is not ill-conditioned. Consider the following example.

Example 1. Consider

$$\begin{array}{ll} \min & f(x) = x^2 + 1 \\ \text{s.t.} & x - 1 = 0. \end{array}$$

The optimal solution is $x^* = 1$. We have:

$$H_1^{\mu}(x) = x^2 + 1 + \mu(x-1)^2,$$

$$\nabla H_1^{\mu}(x) = 2x + 2\mu(x-1),$$

$$\nabla^2 H_1^{\mu}(x) = 2 + 2\mu,$$

and

$$H_2^{\mu}(x) = (x^2+1)/\mu + (x-1)^2,$$

$$\nabla H_2^{\mu}(x) = 2x/\mu + 2(x-1),$$

$$\nabla^2 H_2^{\mu}(x) = 2/\mu + 2.$$

Therefore when $\mu \to \infty$, the hessian matrix $\nabla^2 H_1^{\mu}$ tends to infinity, But $\nabla^2 H_2^{\mu}$ tends to a fixed number.

Although under some assumption the hessian of $\nabla^2 H_2^{\mu}$ is not ill-conditioned but there is a problem. For every feasible point x we have $\nabla P(x) = 0$, and for too large μ, the value of $\nabla f(x)/\mu$ is very close to zero. Thus, near the boundary of feasible region, $\nabla H_2^{\mu} = \nabla\left(f(x)/\mu + P(x)\right)$ is almost zero and this cause the termination of the penalty method. So the penalty method with H_2^{μ} only gives a feasible point and does not converge to optimal solution or converges very slowly.

Thus, to have advantages of both H_1^{μ} and H_2^{μ}, we consider the following combined formula:

$$H_3^{\mu}(x) = f(x)/\mu + \mu P(x).$$

This penalty function apply penalty two times, once by multiplying $P(x)$ by μ and again by dividing $f(x)$ by μ. In fact, H_3^{μ} is equivalent to the following penalty function in which a μ has been factorized:

$$H_4^{\mu}(x) = f(x) + \mu^2 P(x).$$

But order of $\nabla^2 H_3^{\mu}$ is $O(\mu)$ while order of $\nabla^2 H_4^{\mu}$ is $O(\mu^2)$. This leads to faster convergence of penalty method using H_3^{μ} than that using H_4^{μ}.

We use the following general formula instead of H_3^{μ}:

$$H^{\mu}(x) = \frac{f(x)}{\phi(\mu)} + \mu P(x),$$

where, $\phi : \Re \to \Re^+$ is a positive and increasing function in terms of μ.

Lemma 2.1 *Consider the following problem*:

$$(PEN) \qquad \begin{array}{ll} \min & H^{\mu}(x) = \dfrac{f(x)}{\phi(\mu)} + \mu P(x) \\ \text{s.t.} & x \in X. \end{array}$$

Suppose that for each $\mu > 0$ there exists a solution $x^{\mu} \in X$ for (*PEN*), and that x^{μ} is obtained in a compact subsets of X. Then, any limit point of x^{μ} is a solution to (*NLP*).

Proof. Consider the following problem:

$$\begin{array}{ll} \min & f(x) + \mu\phi(\mu) P(x) \\ \text{s.t.} & x \in X, \end{array}$$

Since $f(x)+\mu\phi(\mu)P(x)=\phi(\mu)H^{\mu}(x)$, clearly the problem is equivalent to (*PEN*). Since $\mu\phi(\mu)\to\infty$ when $\mu\to\infty$, thus considering $\mu\phi(\mu)$ as a penalty parameter and applying Theorem 9.2.2 of [6] implies the result.

Although $\nabla^2 H_1^{\mu}$ and $\nabla^2 H^{\mu}$ are both of $O(\mu)$ order, but H_1^{μ} is a penalty function with penalty parameter μ and H^{μ} is equivalent to a penalty function with penalty parameter $\mu\phi(\mu)$ (see proof of Lemma 2.1). Since for larger penalty parameter solution of penalty problem is closer to the solution of main problem, largeness of $\mu\phi(\mu)$ in comparison with μ leads to faster convergence of the penalty method.

3. Extension to Augmented Lagrangian Methods

The augmented Lagrangian for Problem (*NLP*) is defined as follows:

$$A_1^{\mu}(x,\lambda)=f(x)+\sum_{j=1}^{m}\lambda_j G_j^{\mu}+\mu\sum_{j=1}^{m}\left(G_j^{\mu}\right)^2$$

where $G_j^{\mu}=\max\left\{g_j(x),-\frac{\lambda_j}{2\mu}\right\}$.

It has been shown that if λ^* is the Lagrange multiplier of (*NLP*) at the optimal solution x^* Then for large enough μ, minimization of $A_1^{\mu}\left(x,\lambda^*\right)$ gives the optimal solution of (*NLP*). Thus, A_1^{μ} is said to be exact for solving (*NLP*).

Since at first the value of λ^* is not often available, the following formula is usually used for updating the values of λ_j:

$$\lambda_j^{k+1}=\lambda_j^k+2\mu_k G_j^{\mu_k},\qquad j=1,2,\cdots,m$$

Now consider $A_1^{\mu\phi(\mu)}(x,\lambda)$. We can write it as follows:

$$A_1^{\mu\phi(\mu)}(x,\lambda)=\phi(\mu)\left(\frac{f(x)+\sum_{j=1}^{p}\lambda_j G_j^{\mu\phi(\mu)}}{\phi(\mu)}+\mu\sum_{j=1}^{m}\left(G_j^{\mu\phi(\mu)}\right)^2\right)$$

Thus, from the discussion of previous section, instead of A_1^{μ} we consider the following penalty function:

$$A^{\mu}(x,\lambda)=\frac{f(x)+\sum_{j=1}^{p}\lambda_j G_j^{\mu\phi(\mu)}}{\phi(\mu)}+\mu\sum_{j=1}^{m}\left(G_j^{\mu\phi(\mu)}\right)^2$$

Since the ordinary augmented Lagrangian method for solving (*NLP*) is exact and we also have

$$A^{\mu}(x,\lambda)=\frac{1}{\phi(\mu)}A_1^{\mu\phi(\mu)},\qquad (3.1)$$

clearly similar to the ordinary augmented Lagrangian method we have the following result.

Lemma 3.1 *Suppose that second order sufficient conditions for* (*NLP*) *are satisfied at* x^*, λ^*. *Then there exists a* μ_0 *such that for any* $\mu>\mu_0$, x^* *is a local minimizer of* $A^{\mu}\left(x,\lambda^*\right)$.

From (3.1), we can consider A^{μ} as an ordinary augmented lagrangian with penalty parameter $\mu\phi(\mu)$. Thus, new updating formula for the λ_j is as follows:

$$\lambda_j^{k+1}=\lambda_j^k+2\mu_k\phi(\mu_k)G_j^{\mu_k\phi(\mu_k)},\qquad j=1,2,\cdots,m$$

4. Computational Results

4.1. Algorithms

Consider the following augmented Lagrangian problem for (*NLP*):

$$(PA)\quad \min\quad A(x,\lambda)=\frac{f(x)+\sum_{j=1}^{p}\lambda_j G_j^{\mu_j\phi(\bar{\mu})}}{\phi(\bar{\mu})}+\sum_{j=1}^{m}\mu_j\left(G_j^{\mu_j\varphi(\bar{\mu})}\right)^2$$
$$\text{s.t.}\quad x\in X.$$

where, μ is the average of the μ_j. For solving (*NLP*) via augmented Lagrangian method we apply the following algorithm where is similar to Algorithm 1 of [11] with the first order update rule of Lagrangian multipliers.

Algorithm 1

Define $G(x)=\left[G_1^{\mu_1\phi(\bar{\mu})}(x),\cdots,G_m^{\mu_m\phi(\bar{\mu})}(x)\right]^{\mathrm{T}}$.

{Given: x^0,λ^0 }
$x\leftarrow x^0$
$\lambda\leftarrow\lambda^0$
$\mu_j\leftarrow 2,\ j=1,\cdots,m$
$viol^-\leftarrow\left\|G(x)\right\|_\infty$
while $iol>10^{-8}$
{line search method for solving (*PA*)}
$counter\leftarrow 0$
while $\left\|\nabla_x A\right\|>10^{-16}$ and
$counter<3(m+n+1)$
$d_N\leftarrow$ modified BFGS direction
$\theta\leftarrow$ Goldestein stepsize

$x \leftarrow x + \theta d_N$
$counter \leftarrow counter + 1$
end(while)
$viol \leftarrow \|G(x)\|_\infty$
if $viol < 1/4 viol^-$
$$\lambda_j \leftarrow \lambda_j + 2\mu_j \phi(\bar{\mu}) G_j^{\mu_j \phi(\bar{\mu})},\ j = 1,\cdots,m$$
end(if)
for $j = 1,\cdots,m$
if $G_j^{\mu_j \phi(\bar{\mu})}(x) > 1/4 viol^-$
$$\mu_j \leftarrow \left(\mu_j \phi(\bar{\mu})\right)^{1.3} / \phi(\bar{\mu})$$
end(if)
end(for)
if $viol < 1/4 viol^-$
$viol^- \leftarrow viol$
end(if)
end(while)

For solving (*NLP*) via the penalty method, we refine Algorithm 1 by considering λ as zero and removing the step of its updating. Also, we solve the following problem in line search method of the algorithm:

$$\min \quad H(x) = \frac{f(x)}{\phi(\bar{\mu})} + \sum_{j=1}^{m} \mu_j g_j^+(x)^2$$
$$\text{s.t.} \quad x \in X.$$

4.2. Test Results

Algorithms 1 is programmed in MATLAB 7.6 and run on a PC with 1.8 GHz and 1 GB RAM. For solving subproblems we use a line search algorithm. The step length is determined by the Goldstein test and the direction is determined by the BFGS formula with Powell's modifications [16] (the eigenvalues are considered as zero). The function ϕ is considered as $\phi(\mu) = \mu^\alpha$ for $\alpha = 0, \frac{1}{4}, \frac{1}{2}, 1, 1.5, 2, 4$. For each test problem we take a fixed initial point.

All the test problems with one or more constraints are selected from Hock and Schittkowski's set [17] and Schittkowski''s set [18] located in [19]. The characteristics of test problems are listed in **Table 1**, where n is the number of variables, m the total number of constraints, m_{NL} the number of nonlinear constraints and *objective* the type of the objective function (linear/nonlinear).

The computational results for the penalty method and the augmented Lagrangian method are summarized in **Tables 2** and **3**, respectively. The following symbols are used in these tables:

val^* = optimal value of the test problem.
val = the obtained optimal value.
iter= number of iterations.
eval = number of function evaluations.
$eval_0$ = *eval* for the ordinary penalty method ($\alpha = 0$)
$\bar{\mu}$ = the average of $\mu_j, j = 1,\cdots,m$ when the algorithm terminates.
$\bar{\mu}_0 = \bar{\mu}$ in case $\alpha = 0$.
time: CPU time (seconds) to reach the solution.

Table 1. Problem characteristics for test problems.

#	problem	*n*	*m*	m_{NL}	*objective*
1	hs047	5	3	3	nonlinear
2	hs050	5	3	0	nonlinear
3	hs100	7	4	4	nonlinear
4	hs113	10	8	5	nonlinear
5	s216	2	1	1	nonlinear
6	s219	4	2	2	linear
7	s266	15	10	10	nonlinear
8	s385	15	10	10	linear
9	s388	15	25	11	linear
10	s394	20	1	1	nonlinear

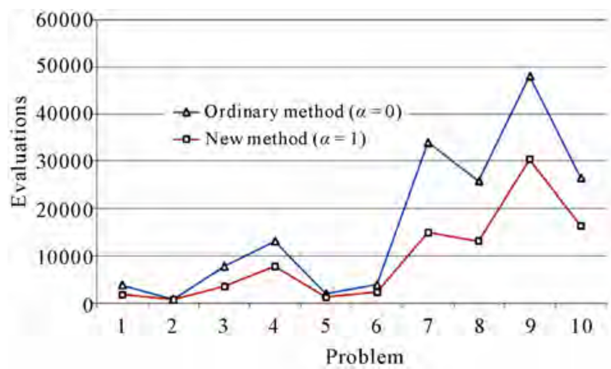

Figure 1. Comparison of ordinary penalty method (*α* = 0) and new penalty method (*α* = 1).

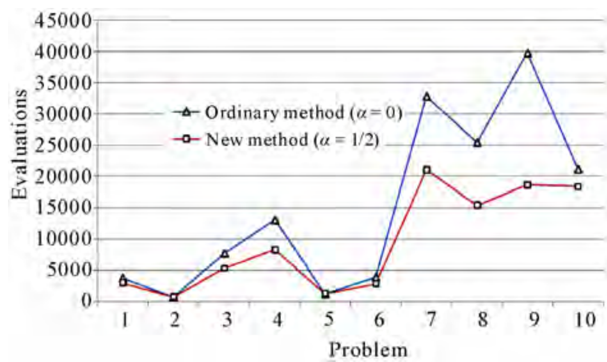

Figure 2. Comparison of ordinary augmented Lagrangian (*α* = 0) and new augmented Lagrangian (*α* = 1/2).

Table 2. Numerical results for the ordinary penalty method (α = 0) and the new penalty method (α = 1, 2).

	α	*iter*	*val*	$\bar{\mu}$	*eval*	*time*	$\bar{\mu}/\bar{\mu}_0$	*eval/eval*$_0$
problem 1	0	10	0	1.41E+04	3813	0.44	1.0E+00	1.00
$val^* = 0$	1	5	0.000 001	1.43E+03	1898	0.22	1.0E−01	0.50
	2	5	0.000 02	2.84E+07	1914	0.22	2.0E+03	0.50
problem 2	0	2	0	3.23E+00	781	0.08	1.0E+00	1.00
$val^* = 0$	1	2	0	5.90E+00	777	0.09	1.8E+00	0.99
	2	2	0	1.22E+01	775	0.09	3.8E+00	0.99
problem 3	0	14	680.630 057	3.56E+11	7781	0.86	1.0E+00	1.00
$val^* = 680.630057$	1	7	680.630 058	2.77E+06	3664	0.41	7.8E−06	0.47
	2	5	680.630 057	2.43E+05	2210	0.25	6.8E−07	0.28
problem 4	0	14	24.306 21	4.45E+11	13 125	1.55	1.0E+00	1.00
$val^* = 24.306209$	1	8	24.306 216	1.53E+07	7871	0.94	3.4E−05	0.60
	2	5	24.306 212	3.60E+05	6426	0.78	8.1E−07	0.49
problem 5	0	21	0.999 375	1.31E+09	1944	0.23	1.0E+00	1.00
$val^* = 0.999375$	1	13	0.999 375	1.12E+05	1286	0.17	8.6E−05	0.66
	2	10	0.999 375	8.38E+03	1055	0.13	6.4E−06	0.54
problem 6	0	21	−1	1.31E+09	3862	0.42	1.0E+00	1.00
$val^* = -1$	1	12	−1	1.12E+05	2351	0.27	8.6E−05	0.61
	2	9	−1	8.38E+03	1798	0.22	6.4E-06	0.47
problem 7	0	37	1	6.59E+08	33 986	8.28	1.0E+00	1.00
$val^* = 1$	1	15	1	1.08E+04	15 048	3.67	1.6E−05	0.44
	2	10	1	2.51E+04	13 069	3.19	3.8E−05	0.38
problem 8	0	20	−8314.945 797	7.07E+18	25 786	4.25	1.0E+00	1.00
$val^* = -8314.945797$	1	9	−8314.945 715	7.07E+18	13 202	2.22	1.0E+00	0.51
	2	5	−8314.945 753	1.54E+06	11 214	1.89	2.2E−13	0.43
problem 9	0	30	−5821.084 223	4.72E+08	48 089	12.89	1.0E+00	1.00
$val^* = -5821.084225$	1	21	−5821.084 224	1.06E+05	30 418	8.20	2.2E−04	0.63
	2	12	−5821.084 218	8.46E+02	18 615	5.00	1.8E−06	0.39
problem 10	0	21	1.916 667	1.31E+09	26 446	2.70	1.0E+00	1.00
$val^* = 1.916667$	1	12	1.916 667	1.12E+05	16437	1.70	8.6E−05	0.62
	2	9	1.916 667	8.38E+03	17694	1.81	6.4E−06	0.67

Note that tables rows corresponding to $\alpha = 0$ show numerical results for the ordinary methods and other rows show numerical results for the new methods.

As seen in **Tables 2** and **3** the performance of the penalty methods with new formulas is significantly better than that with the usual formulas. The new penalty me-

Table 3. Numerical results for the ordinary penalty method (α = 0) and the new penalty method (α = 1/2, 1).

	α	*iter*	*val*	$\bar{\mu}$	*eval*	*time*	$\bar{\mu}/\bar{\mu}_0$	*eval/eval$_0$*
problem 1	0	10	0	1.41E+04	3813	0.42	1.0E+00	1.00
$val^* = 0$	1/2	8	0	7.63E+05	3041	0.33	5.4E+01	0.80
	1	5	0.000 001	1.43E+03	1898	0.22	1.0E−01	0.50
problem 2	0	2	0	3.23E+00	781	0.08	1.0E+00	1.00
$val^* = 0$	1/2	2	0	4.29E+00	783	0.08	1.3E+00	1.00
	1	2	0	5.90E+00	777	0.09	1.8E+00	0.99
problem 3	0	14	680.630 057	3.56E+11	7781	0.89	1.0E+00	1.00
$val^* = 680.630057$	1/2	10	680.630 058	1.64E+10	5355	0.61	4.6E−02	0.69
	1	7	680.630 058	2.77E+06	3664	0.42	7.8E−06	0.47
problem 4	0	14	24.306 21	4.45E+11	13 125	1.55	1.0E+00	1.00
$val^* = 24.306209$	1/2	9	24.306 209	7.36E+07	8376	0.98	1.7E−04	0.64
	1	8	24.306 216	1.53E+07	7871	0.91	3.4E−05	0.60
problem 5	0	14	0.999 375	7.74E+01	1269	0.16	1.0E+00	1.00
$val^* = 0.999375$	1/2	17	0.999 379	3.38E+08	1346	0.14	4.4E+06	1.06
	1	11	0.999 375	1.71E+01	994	0.11	2.2E−01	0.78
problem 6	0	19	−1	2.85E+02	3928	0.45	1.0E+00	1.00
$val^* = -1$	1/2	13	−1	2.24E+01	2908	0.33	7.8E−02	0.74
	1	14	−1	1.14E+02	3125	0.36	4.0E−01	0.80
problem 7	0	40	1	1.34E+02	32 887	7.66	1.0E+00	1.00
$val^* = 1$	1/2	21	1	3.88E+01	21 026	4.94	2.9E-01	0.64
	1	22	1	1.33E+02	17 851	4.19	1.0E+00	0.54
problem 8	0	18	−8314.945 797	7.07E+16	25 476	4.19	1.0E+00	1.00
$val^* = -8314.945797$	1/2	9	−8314.945 793	1.06E+08	15 430	2.58	1.5E-09	0.61
	1	9	−8314.945 715	7.07E+18	13 202	2.22	1.0E+02	0.52
problem 9	0	36	−5821.084 218	1.10E+04	39 826	10.52	1.0E+00	1.00
$val^* = -5821.084225$	1/2	15	−5821.084 18	3.21E+00	18 624	5.00	2.9E−04	0.47
	1	26	−5821.084 224	2.93E+03	30 457	8.03	2.7E−01	0.76
problem 10	0	17	1.9166 67	2.85E+02	21 168	2.23	1.0E+00	1.00
$val^* = 1.916667$	1/2	15	1.9166 67	1.14E+04	18 404	1.91	4.0E+01	0.87
	1	12	1.9168 88	1.12E+05	10 483	1.09	3.9E+02	0.50

thods decrease number of iterations and number of function evaluations and as we expect the penalty method notably reduce the penalty parameter.

We observed in computational results that although for larger α the convergence is faster, for some test problems use of larger α increase the distance of the obtained solution and the optimal solution. Note that using $\alpha > 1$ sometimes makes the first term of H^{μ} converges to zero faster than the second term and this causes termination of the penalty method at the boundary of feasible region. Thus, we suggest use of μ^{α} such that $\alpha \le 1$. That is, use of the ϕ with the order greater than $O(\mu)$ is not recommended. Here, for having more efficiency we suggest $\phi(\mu) = \mu$ for the penalty method and $\phi(\mu) = \sqrt{\mu}$ for the augmented Lagrangian method.

In **Figure 1** number of function evaluations for the ordinary penalty method ($\alpha = 0$) and new penalty method ($\alpha = 1$) is compared. The comparison of evaluations of ordinary augmented Lagrangian ($\alpha = 0$) and new augmented Lagrangian ($\alpha = 1/2$) is illustrated in **Figure 2**.

5. Conclusions

We proposed a simple modification to the penalty methods and showed that the new penalty methods has better performance than the usual penalty methods. Computational results on several test problems showed that number of iterations decreases and calculations significantly reduce.

6. References

[1] R. Courant, "Variational Methods for the Solution of Problems of Equilibrium and Vibrations," *Bulletin of the American Mathematical Sociaty*, Vol. 49, No. 1, 1943, pp. 1-23. doi:10.1090/S0002-9904-1943-07818-4

[2] A. V. Fiacco and G. P. McCormick, "Nonlinear Programming: Sequential Unconstrained Minimization Techniques," Society for Industrial and Applied Mathematics, McLean, 1990.

[3] D. M. Murray and S. J. Yakowitz, "Constrained Differential Dynamic Programming and Its Application to Multireservior Control," *Water Resources Research*, Vol. 15, No. 5, 1979, pp. 1017-1027. doi:10.1029/WR015i005p01017

[4] W. I. Zangwill, "Nonlinear Programming via Penalty Functions," *Management Science*, Vol. 13, No. 5, 1967, pp. 344-358. doi:10.1287/mnsc.13.5.344

[5] R. Fletcher, "A Class of Methods for Nonlinear Programming with Termination and Convergence Properties," Integer and Nonlinear Programming, Amsterdam, 1970, pp. 157-173.

[6] M. S. Bazaraa, H. D. Sherali and C. M. Shetty, "Nonlinear Programming: Theory and Algorithms," 3rd Edition, Wiley, New York, 2006.

[7] C. Charalambous, "A Lower Bound for the Controlling Parameters of the Exact Penalty Functions," *Mathematical Programming*, Vol. 15, No. 1, 1978, pp. 278-290. doi:10.1007/BF01609033

[8] A. R. Conn, "Constrained Optimization Using a Nondifferentiable Penalty Function," *SIAM Journal of Numerical Analysis*, Vol. 10, No. 4, 1973, pp. 760-784. doi:10.1137/0710063

[9] G. D. Pillo and L. Grippo, "A Continuously Differentiable Exact Penalty Function for Nonlinear Programming Problems with Inequality Constraints," *SIAM Journal of Control and Optimization*, Vol. 23, No. 1, 1985, pp. 72-84. doi:10.1137/0323007

[10] G. D. Pillo and L. Grippo, "Exact Penalty Functions in Constrained Optimization," *SIAM Journal of Control and Optimization*, Vol. 27, No. 6, 1989, pp. 1333-1360. doi:10.1137/0327068

[11] J.-P. Dussault, "Improved Convergence Order for Augmented Penalty Algorithms," *Computational Optimization and Applications*, Vol. 44, No. 3, 2009, pp. 373-383. doi:10.1007/s10589-007-9159-0

[12] A. L. Peressini, F. E. Sullivan and J. J. Uhl, "The Mathematics of Nonlinear Programming," Springer-Verlag, New York, 1988.

[13] T. Pietrzykowski, "An Exact Potential Method for Constrained Maxima," *SIAM Journal of Numerical Analysis*, Vol. 6, No. 2, 1969, pp. 294-304. doi:10.1137/0706028

[14] M. Mongeau and A. Sartenaer, "Automatic Decrease of the Penalty Parameterin Exact Penalty Function Methods," *European Journal of Operational Research*, Vol. 83, No. 3, 1995, pp. 686-699. doi:10.1016/0377-2217(93)E0339-Y

[15] D. G. Luenberger and Y. Ye, "Linear and Nonlinear Programming," 3rd Edition, Springer, New York, 2008.

[16] M. J. D. Powell, "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations," *Lecture Notes in Mathematics*, Vol. 630, 1978, pp. 144-157.

[17] W. Hock and K. Schittkowski, "Test Examples for Nonlinear Programming Codes," *Journal of Optimization Theory and Applications*, Vol. 30, No. 1, 1980, pp. 127-129.

[18] K. Schittkowski, "More Test Examples for Nonlinear Programming Codes (Lecture Notes in Economics and Mathematical Systems)," Springer, Berlin, 1987.

[19] Princeton Library of Nonlinear Programming Models, 2011. http://www.gamsworld.org/performance/princetonlib/princetonlib.htm

