```
function [xk,trace, status] = gd_bt(f, g, x0, s, alpha, beta,
 max_iters, epsilon)
trace = zeros(max_iters, 1);
xk = x0;

for k = 1:max_iters
    trace(k) = f(xk);
    % Determine new step size
    dk = -g(xk); % Negative gradient is descent direction
%     fprintf("k = %3d norm_grad = %2.6f f_val = %2.6f\n", k,
 norm(dk), f(xk));

    tk = s;
    for i = 0:20
        if f(xk) - f(xk + tk*dk) >= -alpha*tk*g(xk)'*dk
            break
        end
        tk = s*beta^i;
    end

    xk = xk + tk*dk;
    % Early stopping conditions
    if norm(g(xk)) < epsilon
        trace = trace(1:k);
        status = 1;
        return
%     elseif k > 2 && trace(k) > trace(k-1)
%         trace = trace(1:k);
%         status = -1;
%         return
    elseif isnan(norm(xk)) || ~isfinite(norm(xk))
        trace = trace(1:k);
        status = -1;
        return
    end
end
status = 0;
end
```

Published with MATLAB® R2018a