

```
function out = optimize (pr)
```

```
(check input)
```

```
(set default values)
```

```
(initialize output variable)
```

```
(initialize main iteration)
```

```
(set up terminal output)
```

```
goflag = true
```

```
while goflag
```

```
    switch pr.method
```

```
        case 'GD'
```

```
            (p = -∇f)
```

```
            (call linesearch)
```

```
        case 'CG'
```

```
            • • •
```

```
        case 'BFGS'
```

```
            • • •
```

```
        case 'TR'
```

```
            • • •
```

```
    end
```

```
(updates)
```

```
(check termination criteria)
```

```
end
```

```
function [pr,msg] = setdefaults(pr)
```

```
msg = { };
```

```
df.dxtol = 1E-8;
```

```
df.c1 = 0.001;
```

```
⋮
```

```
fn = fieldnames(df);
```

```
for k = 1:length(fn)
```

```
    if ~isfield(pr, fn{k}) || isempty(pr.(fn{k}))
```

```
        pr.(fn{k}) = df.(fn{k});
```

```
    end
```

```
end
```

```
(set any messages)
```

```
return
```

```
def SetDefaults(alg):
```

```
    import numpy as np
```

```
    alg.setdefault('dxtol', 1E-8)
```

```
    alg.setdefault('ci', 0.001)
```

```
    :
```

```
    (set any return messages)
```

```
    return alg, msg
```

```
function [xnew, flag] = linesearch ( ... )  
switch pr.linesearch  
    case 'Armijo'  
        (backtracking algorithm 3.1)  
    case 'StrongWolfe'  
        (algorithm 3.5 and subalgorithm 3.6)  
end.
```

```
def LineSearch ( ... )  
    match alg['linesearch']:  
        case 'Armijo':  
            (backtracking algorithm 3.1)  
        case 'StrongWolfe':  
            (algorithm 3.5 and subalgorithm 3.6)  
    return xnew, flag
```