```
function [x, y, chain] = returnrandsearch(this, eps)

    xk = this.m_x0;

    l = this.L0;


    k = 0;

    j = 1;

    fk = this.f(xk);

    this.addpointtochain(xk);


    while true

        r = this.rand();

        xNew = xk + l * r / sqrt(sum(r.*r));

        fNew = this.f(xNew);


        if (fNew < fk)

            xk = xNew;

            fk = fNew;

            k = k + 1;

            this.addpointtochain(xk);

            if (k > this.rand_n)

                x = xk;

                y = fk;

                return;

            end

            j = 0;

        else

            if (j < this.rand_m)

                j = j + 1;

            else
```

```
                    if (l < eps)
                        x = xk;
                        y = fk;
                        return;
                    else
                        l = l * this.delta;
                        j = 1;
                    end
                end
            end
        end
    end

    function [r] = rand(this)
        r = 2*randn(1, this.N) - ones(1, this.N);
    end

    function addpointtochain(this, xk)
        this.m_chaini = this.m_chaini + 1;
        this.m_chain(this.m_chaini, :) = xk;
    end

    function [x, y] = fminsearch(this, eps)
        [x, y] = fminsearch(@this.f, this.m_x0, optimset('TolX', eps));
    end
    end
```