Branch: master ▾ hobby / GA / cma-es.m

Find file | Copy path

koji-endo add indent 23dd8d6 3 minutes ago

1 contributor

64 lines (64 sloc) | 2.55 KB

```
% CMA-ES for non-linear function minimization
% See also http://www.bionik.tu-berlin.de/user/niko
function xmin=cmaes
	% Set dimension, fitness fct, stop criteria, start values...
	N=10; strfitnessfct = 'cigar';
	maxeval = 300*(N+2)^ 2; stopfitness = 1e-10; % stop criteria
	xmeanw = ones(N, 1); % object parameter start point (weighted mean)
	sigma = 1.0; minsigma = 1e-15; % step size, minimal step size
	% Parameter setting: selection,
	lambda = 4 + floor(3*log(N)); mu = floor(lambda/2);
	arweights = log((lambda+1)/2) - log(1:mu)'; % for recombination
	% parameter setting: adaptation
	cc = 4/(N+4); ccov = 2/(N+2^ 0.5)^ 2;
	cs = 4/(N+4); damp = 1/cs + 1;
	% Initialize dynamic strategy parameters and constants
	B = eye(N); D = eye(N); BD = B*D; C = BD*transpose(BD);
	pc = zeros(N,1); ps = zeros(N,1);
	cw = sum(arweights)/norm(arweights);
	chiN = N^ 0.5*(1-1/(4*N)+1/(21*N^ 2));
	% Generation loop
	counteval = 0; arfitness(1) = 2*abs(stopfitness)+1;
	while arfitness(1) > stopfitness & counteval < maxeval
		for k=1:lambda
			% repeat the next two lines until arx(:,k) is feasible
			arz(:,k) = randn(N,1);
			arx(:,k) = xmeanw + sigma * (BD * arz(:,k)); % Eq.(13)
			arfitness(k) = feval(strfitnessfct, arx(:,k));
		counteval = counteval+1;
		end
		% Sort by fitness and compute weighted mean
		[arfitness, arindex] = sort(arfitness); % minimization
		xmeanw = arx(:,arindex(1:mu))*arweights/sum(arweights);
		zmeanw = arz(:,arindex(1:mu))*arweights/sum(arweights);
		% Adapt covariance matrix
		pc = (1-cc)*pc + (sqrt(cc*(2-cc))*cw) * (BD*zmeanw); % Eq.(14)
		C = (1-ccov)*C + ccov*pc*transpose(pc); % Eq.(15)
		% adapt sigma
		ps = (1-cs)*ps + (sqrt(cs*(2-cs))*cw) * (B*zmeanw); % Eq.(16)
		sigma = sigma * exp((norm(ps)-chiN)/chiN/damp); % Eq.(17)
		% Update B and D from C
		if mod(counteval/lambda, N/10) < 1
		C=triu(C)+transpose(triu(C,1)); % enforce symmetry
		[B,D] = eig(C);
		% limit condition of C to 1e14 + 1
			if max(diag(D)) > 1e14*min(diag(D))
			tmp = max(diag(D))/1e14 - min(diag(D));
			C = C + tmp*eye(N); D = D + tmp*eye(N);
			end
		D = diag(sqrt(diag(D))); % D contains standard
		deviations now
		BD = B*D; % for speed up only
		end % if mod
		% Adjust minimal step size
		if sigma*min(diag(D)) < minsigma ...
		| arfitness(1) == arfitness(min(mu+1,lambda)) ...
		| xmeanw == xmeanw ...
		+ 0.2*sigma*BD(:,1+floor(mod(counteval/lambda,N)))
		sigma = 1.4*sigma;
		end
	end % while, end generation loop
```

```
        disp([num2str(counteval) ': ' num2str(arfitness(1))]);
        xmin = arx(:, arindex(1)); % return best point of last generation
function f=cigar(x)
        f = x(1)^ 2 + 1e6*sum(x(2:end).^ 2);
```