Differential Evolution Algorithm with a Modified Archiving-based Adaptive Tradeoff Model for Optimal Power Flow

Outline

- Search Engine
- Constraint Handling Technique
- Test Cases and Statistical Results

Roots of Differential Evolution

- Developed in 1995 by Rainer Storn and Kenneth Price as a continuous optimization method
- Main idea: use a mutation/recombination operator based on difference(s) between pairs of elements
- Similarities with older direct search methods:
  - Pattern search (Hooke-Jeeves, 1961)
  - Simplex methods (Nelder-Mead,1965)
- Other population based methods involving differences:
  - Particle Swarm Optimization (Kennedy & Eberhart, 1995)

DE webpage http://www.icsi.berkeley.edu/ storn/code.html

Books:

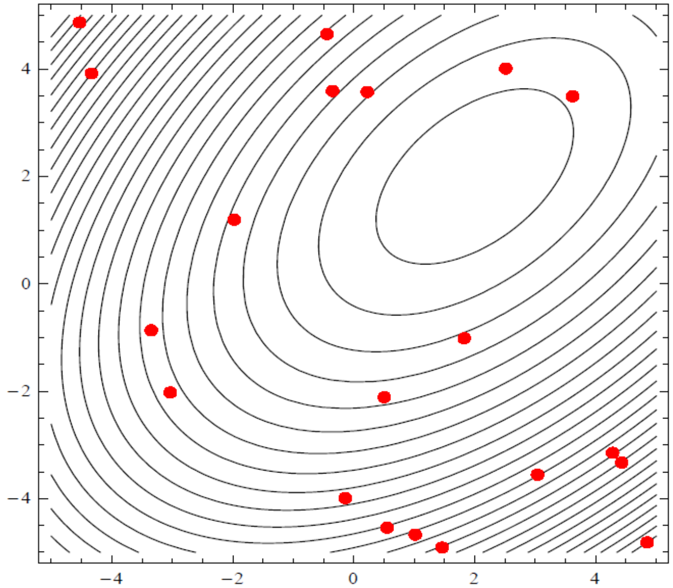
K.V. Price, R.M. Storn, J.A. Lampinen; Differential Evolution. A Practical Approach to Global Optimization, 2005

U. Chakraborty, Advances in Differential Evolution, 2008

Standard Differential Evolution

Problem to be solved: minimize: $f(\vec{x}), \ \vec{x} = (x_1, \ldots, x_D) \in \Re^D$

Initialization: $\{\vec{x}_{i,0} = (x_{i,1,0}, x_{i,2,0}, \ldots, x_{i,D,0}), x_{i,j,0} = U(lb_j, ub_j), i = 1,2,\ldots, NP, \ j = 1,2,\ldots D\}$



i – population size

j – dimensionality

F – scale factor

CR – crossover rate

Standard Differential Evolution

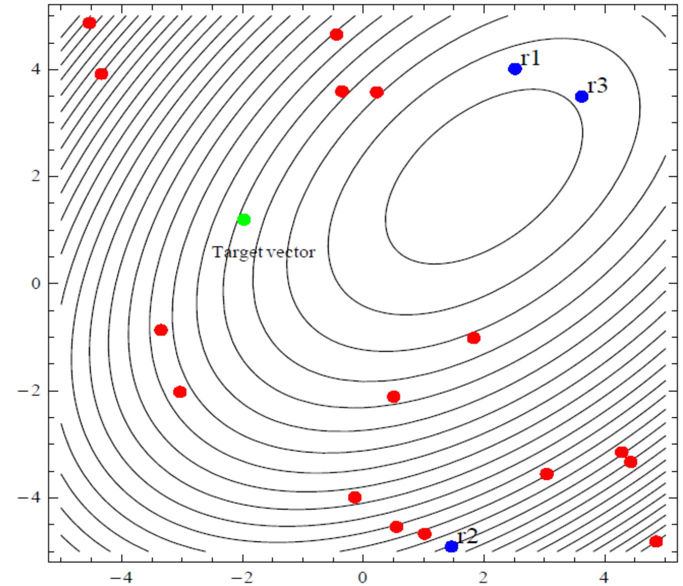
Problem to be solved: minimize: $f(\vec{x}),\ \vec{x} = (x_1, \ldots, x_D) \in \Re^D$

Initialization: $\{\vec{x}_{i,0} = (x_{i,1,0}, x_{i,2,0}, \ldots, x_{i,D,0}),\ x_{i,j,0} = U(lb_j, ub_j),\ i = 1,2,\ldots,NP,\ j = 1,2,\ldots D\}$

while ‹NOT termination› do

•Mutation:

$$\vec{v}_i = \vec{x}_{r_1} + F \cdot (\vec{x}_{r_2} - \vec{x}_{r_3}),\ i = 1,2,\ldots NP$$



i – population size
j – dimensionality
F – scale factor
CR – crossover rate

Standard Differential Evolution

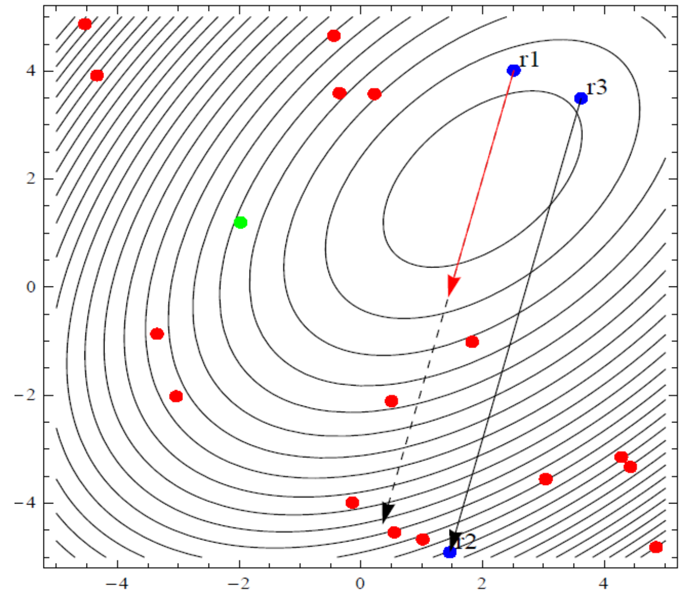
Problem to be solved: minimize: $f(\vec{x}),\ \vec{x}=(x_1,\ldots,x_D)\in\Re^D$

Initialization: $\{\vec{x}_{i,0}=(x_{i,1,0},x_{i,2,0},\ldots,x_{i,D,0}),\ x_{i,j,0}=U(lb_j,ub_j),\ i=1,2,\ldots,NP,\ j=1,2,\ldots D\}$

while ‹NOT termination› do

•Mutation:

$$\vec{v}_i=\vec{x}_{r_1}+F\cdot(\vec{x}_{r_2}-\vec{x}_{r_3}),\ i=1,2,\ldots NP$$



i – population size

j – dimensionality

F – scale factor

CR – crossover rate

Standard Differential Evolution

Problem to be solved: minimize: $f(\vec{x}),\ \vec{x} = (x_1, \dots, x_D) \in \Re^D$

Initialization: $\{\vec{x}_{i,0} = (x_{i,1,0}, x_{i,2,0}, \dots, x_{i,D,0}), x_{i,j,0} = U(lb_j, ub_j), i = 1,2,\dots, NP,\ j = 1,2,\dots D\}$

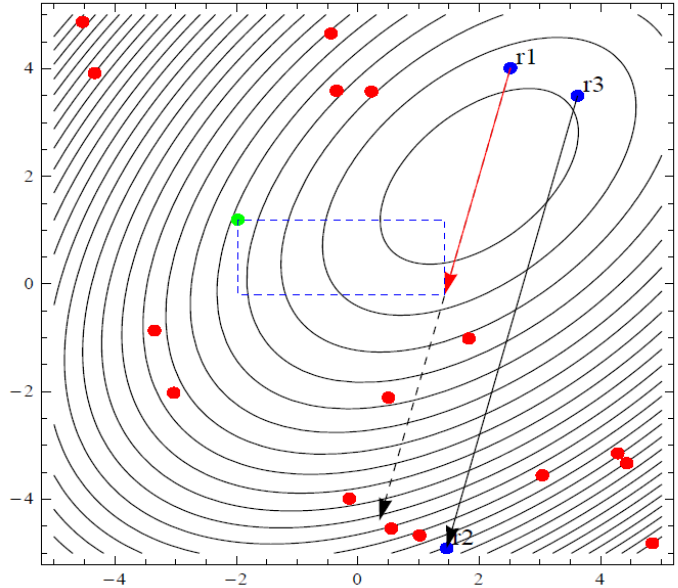
while ‹NOT termination› do

•Mutation:

$$\vec{v}_i = \vec{x}_{r_1} + F \cdot (\vec{x}_{r_2} - \vec{x}_{r_3}),\ i = 1,2,\dots NP$$

•Crossover:

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } rand(0,1) < CR \text{ or } j = j_0 \\ x_{i,j} & \text{otherwise} \end{cases}$$



i – population size

j – dimensionality

F – scale factor

CR – crossover rate

Standard Differential Evolution

Problem to be solved: minimize: $f(\vec{x}),\ \vec{x} = (x_1, \ldots, x_D) \in \Re^D$

Initialization: $\{\vec{x}_{i,0} = (x_{i,1,0}, x_{i,2,0}, \ldots, x_{i,D,0}), x_{i,j,0} = U(lb_j, ub_j), i = 1,2,\ldots,NP,\ j = 1,2,\ldots D\}$

while ‹NOT termination› do

•Mutation:

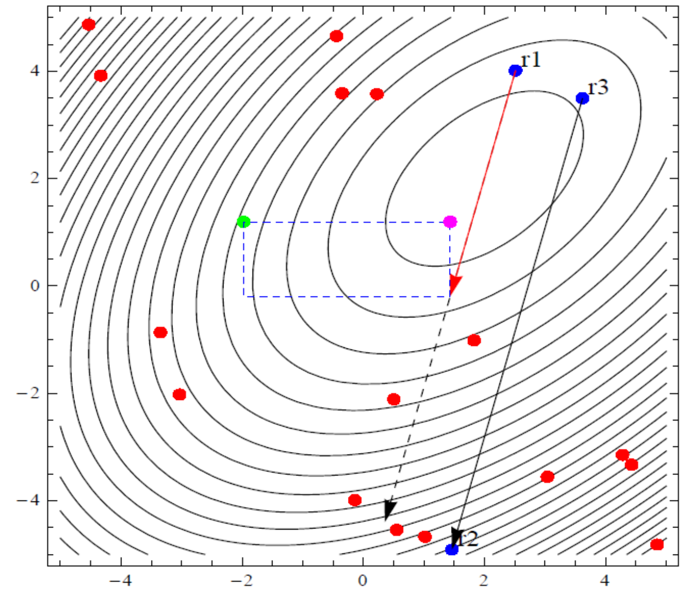
$$\vec{v}_i = \vec{x}_{r_1} + F \cdot (\vec{x}_{r_2} - \vec{x}_{r_3}),\ i = 1,2,\ldots NP$$

•Crossover:

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } rand(0,1) < CR \text{ or } j = j_0 \\ x_{i,j} & \text{otherwise} \end{cases}$$

•Selection:

$$\vec{x}_{i,g+1} = \begin{cases} \vec{u}_i & \text{if } f(\vec{u}_i) \le f(\vec{x}_{i,g}) \\ \vec{x}_{i,g} & \text{if } f(\vec{u}_i) > f(\vec{x}_{i,g}) \end{cases}$$



i – population size
j – dimensionality
F – scale factor
CR – crossover rate

Flexibility of DE

DE taxonomy: DE/base element/ no. of differences/ crossover type

- Base element:
  - random (x_{rand}): DE/rand/*/*
  - best (x_{best}): DE/best/*/*
  - Combination of current and best elements (λx_{best} + (1- λ)x_i):
    DE/current-to-best/*/*
  - combination of random and best elements (λx_{r1} + (1- λ)x_i):
    DE/current-to-rand/*/*
- Number of differences: usually 1 (DE/*/1/*) or 2 (DE/*/2/*)
- Crossover type: binomial (DE/*/*/bin) or exponential (DE/*/*/exp)

Flexibility of DE

The other face of flexibility: which variant to choose ?

Recommendations

- No specific knowledge on the problem: use DE/rand/1/*
- Need for an exploitative method: use DE/best/1/*
- Need for a more explorative method: use DE/rand/2/*

Remark: different variants could be appropriate in different stages of the optimization process

Books:

K.V. Price, R.M. Storn, J.A. Lampinen; Differential Evolution. A Practical Approach to Global Optimization, 2005

U. Chakraborty, Advances in Differential Evolution, 2008

(1+3) DE

1 parent individual generates 3 offspring individuals by 3 different mutation strategies:

DE/rand/1:

$$\vec{v}_i = \vec{x}_{r_1} + F \cdot (\vec{x}_{r_2} - \vec{x}_{r_3})$$

DE/rand/2:

$$\vec{v}_i = \vec{x}_{r_1} + F \cdot (\vec{x}_{r_2} - \vec{x}_{r_3}) + F \cdot (\vec{x}_{r_4} - \vec{x}_{r_5})$$

DE/current-to-best(rand)/1:

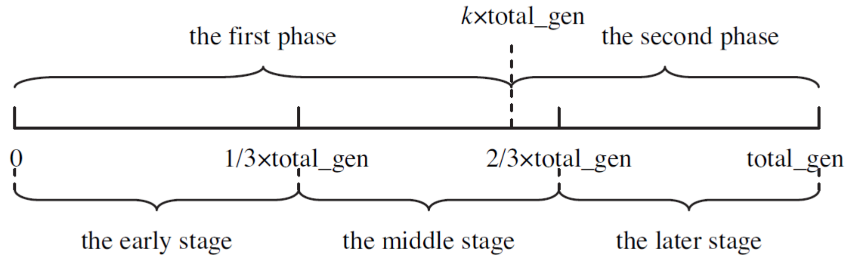
$$\vec{v}_i = \vec{x}_i + F \cdot (\vec{x}_{best} - \vec{x}_i) + F \cdot (\vec{x}_{r_1} - \vec{x}_{r_2}) \qquad \vec{v}_i = \vec{x}_i + F \cdot (\vec{x}_{r_1} - \vec{x}_i) + F \cdot (\vec{x}_{r_2} - \vec{x}_{r_3})$$

(1+3) DE

The current generation number (denoted as ***current_gen***) is compared with a threshold generation number (denoted as ***threshold_gen***)

$$threshold\_gen = k \times total\_gen$$

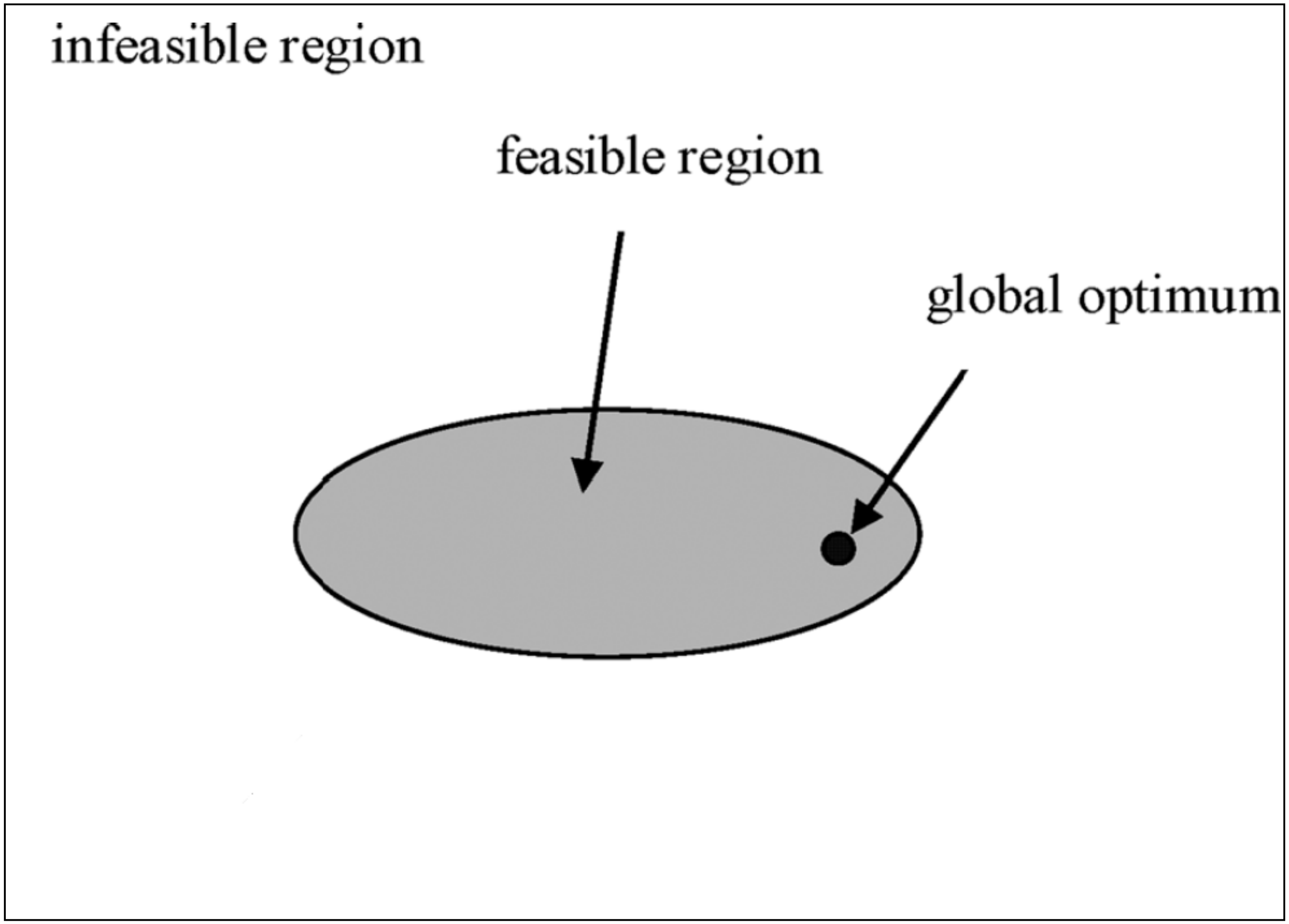
If ***current_gen*** > ***threshold_gen***, the “DE/current-to-best/1” is activated



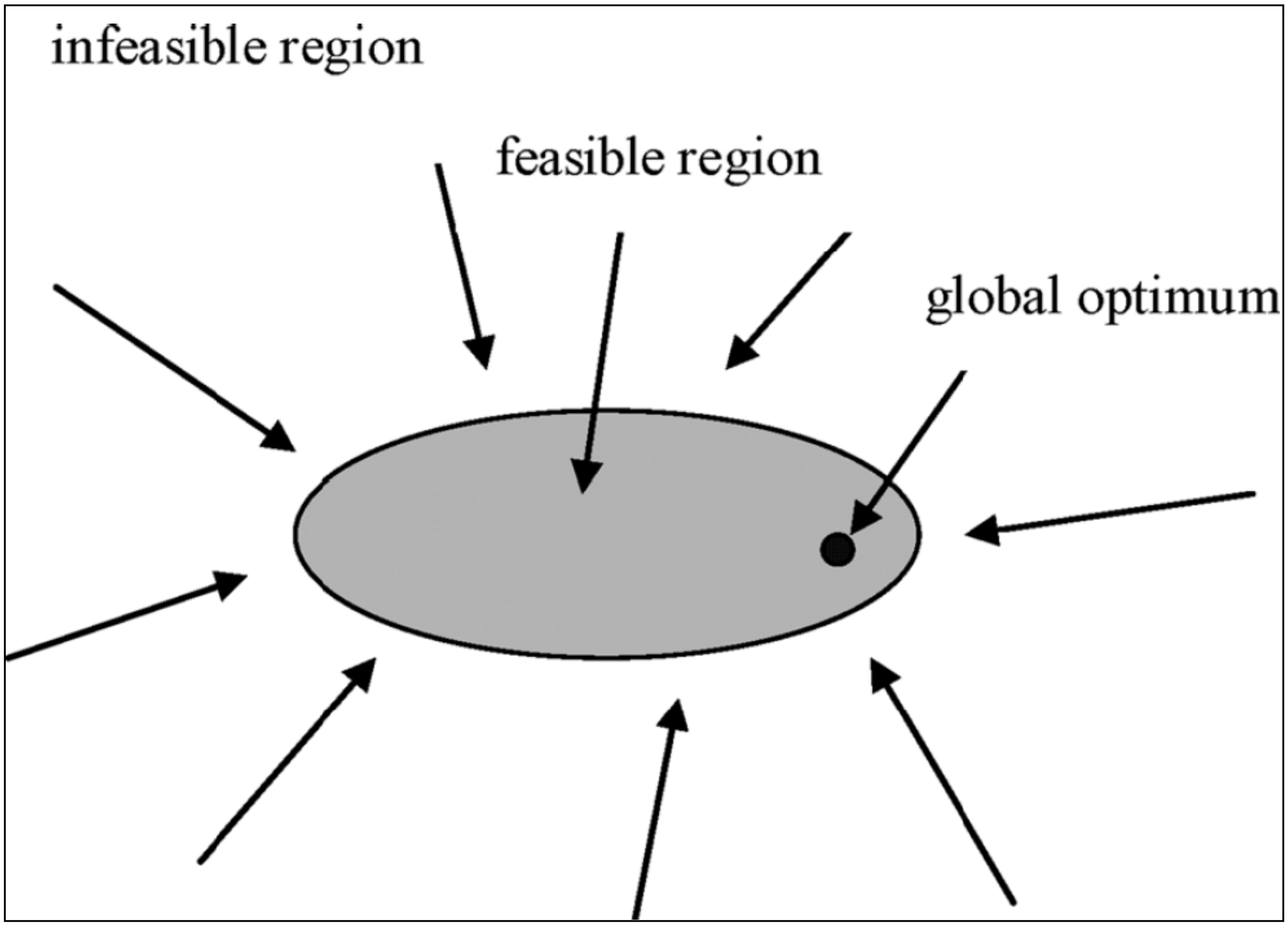
Journal:

G. Jia, Y. Wang, Z. Cai, and Y. Jin. An improved (μ+λ)-constrained differential evolution for constrained optimization. Information Sciences, vol. 222, pp. 302-322, 2013.

Constrained Optimization



Constrained Optimization



Constrained Optimization

In general, for constraint optimization, the population may inevitably experience three situations:

- The population contains infeasible individuals only (infeasible situation)
- The population consists of both feasible and infeasible individuals (semi-feasible situation)
- The population is entirely composed of feasible individuals (feasible situation)

CH for Infeasible Situation

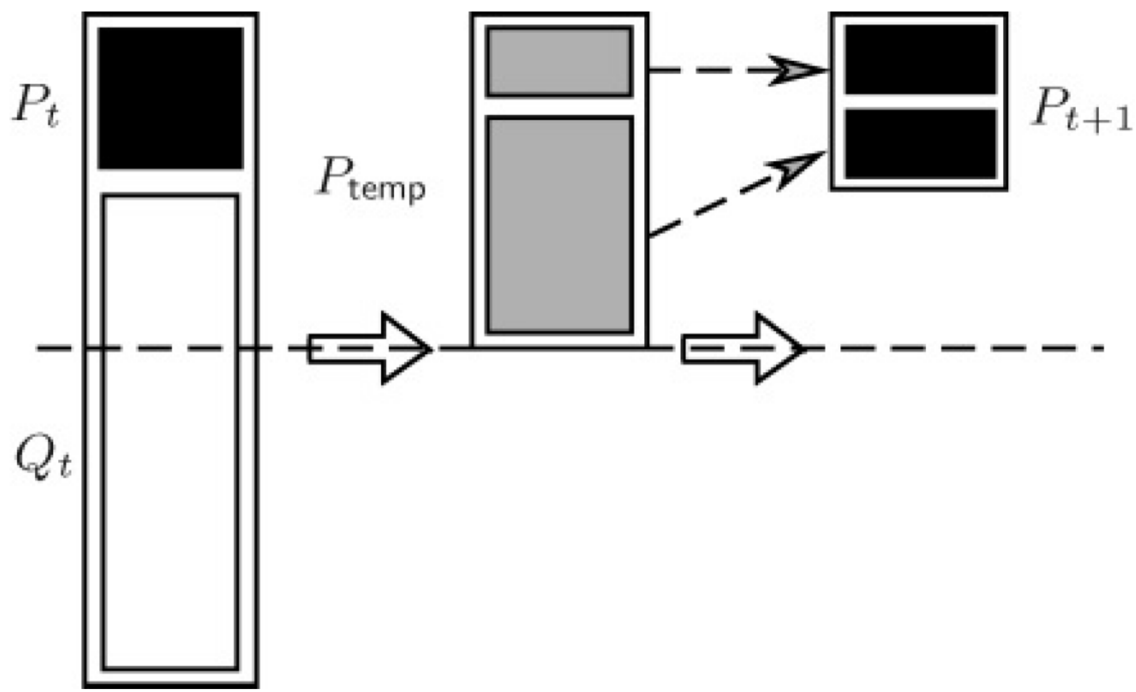
Goals:

- Promptly motivate the population toward the feasible region
- The diversity in the population should be kept

Solutions:

- Truncation technique
- Multiobjective optimization-based technique

Truncation Technique



Journal:

Y. Wang and Z. Cai. Constrained evolutionary optimization by means of (μ+λ)-differential evolution and improved adaptive trade-off model. Evolutionary Computation, vol. 19, no. 2, pp. 249-285, 2011.

Multiobjective optimization-based technique

A MOP can be formulated as follows:

$$\text{minimize } \vec{f}(\vec{x}) = (f_1(\vec{x}),\ldots f_m(\vec{x}))$$

where m is the number of objective functions, $\vec{x} = (x_1,\ldots x_n) \in S \subset \Re^n$ is the decision vector which contains n decision variables and S is the search space.

Definition 1. (*Pareto dominance*): Consider two decision vectors $\vec{a} = (a_1,\ldots a_n)$ and $\vec{b} = (b_1,\ldots b_n)$, ***a*** is said to Pareto dominate ***b*** (denoted as $\vec{a} \prec \vec{b}$), if and only if

$$\forall i \in \{1,\ldots m\},\ f_i(\vec{a}) \le f_i(\vec{b})\ \ and\ \ \exists j \in \{1,\ldots m\}, f_i(\vec{a}) < f_i(\vec{b})$$

In this case, ***b*** is said to be Pareto dominated by ***a***. If Pareto dominance does not hold between ***a*** and ***b***, they are considered nondominated with each other.

Multiobjective optimization-based technique

Definition 2. (*Pareto optimality*): $\vec{a} \in S$ is said to be Pareto optimal in *S*, if and only if $\neg\exists \vec{b} \in S$ satisfies $\vec{b} \prec \vec{a}$.

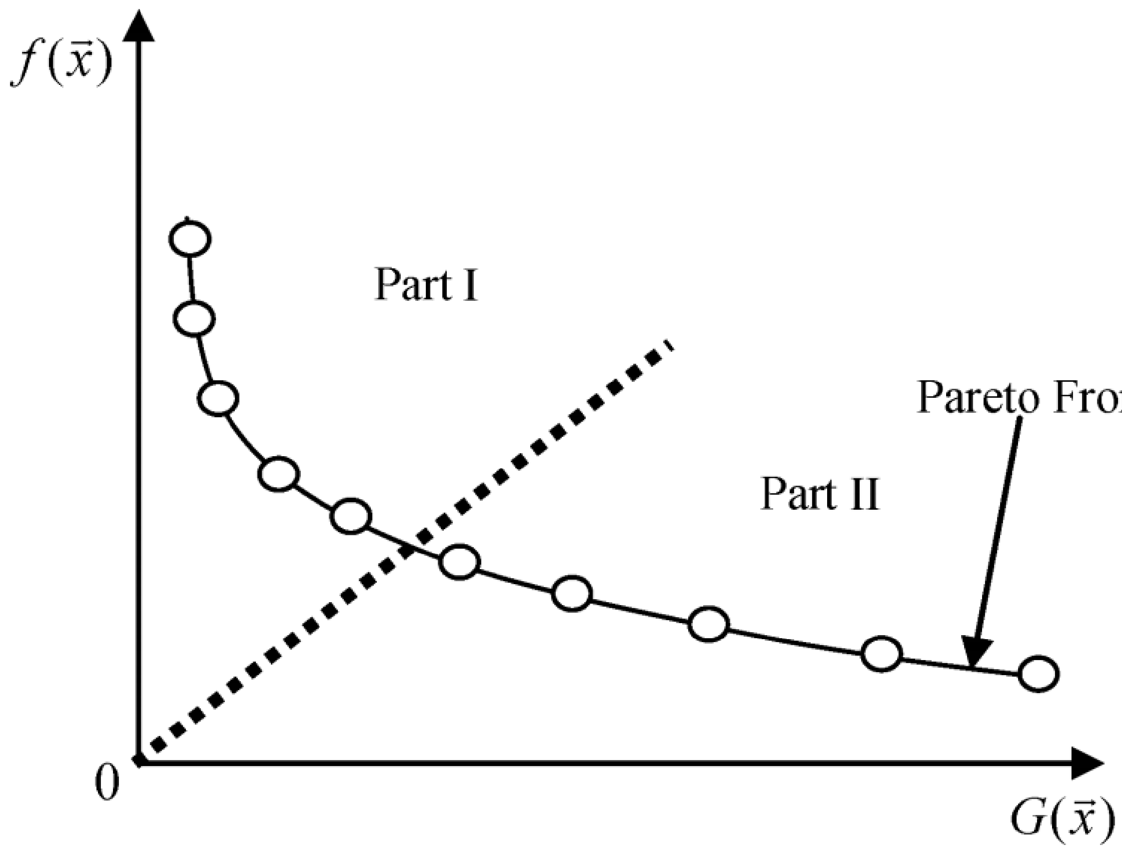
Definition 3. (*Pareto optimal set*): The Pareto optimal set (denoted as $P*$) is the set of all the Pareto optimal solutions:

$$P* = \{\vec{a} \in S \mid \neg\exists \vec{b} \in S, \vec{b} \prec \vec{a}\}$$

Definition 4. (*Pareto front*): The Pareto front (denoted as $PF*$) is defined as:

$$PF* = \{\vec{f}(\vec{a}) \mid \vec{a} \in P*\}$$

Multiobjective optimization-based technique



CH for Semi-feasible Situation

Goals:

- Feasible individuals with small objective function values should remain in the next population
- Infeasible individuals with slight constraint violations and small objective function values should remain in the next population

Solution:

- An adaptive fitness transformation

CH for Semi-feasible Situation

Feasible group: $Z_1 = \{i \mid G(\vec{x}_i = 0\}$

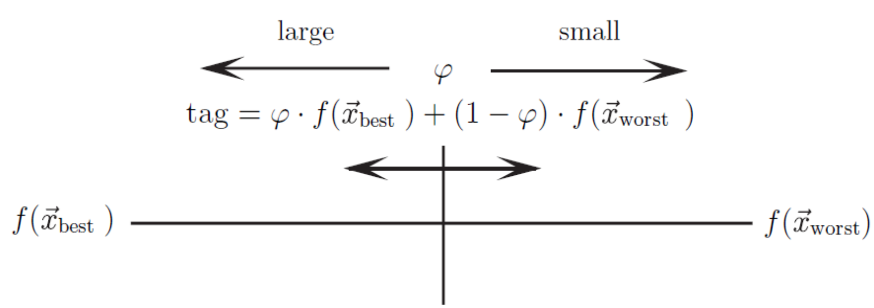
Infeasible group: $Z_2 = \{i \mid G(\vec{x}_i > 0\}$

The best and worst feasible solutions (denoted as $\vec{x}_{best}$ and $\vec{x}_{worst}$) are found from the feasible group.

The converted objective function value:

$$f(\vec{x}_i) = \begin{cases} f(\vec{x}_i), & i \in Z_1 \\ \max\{\varphi \cdot f(\vec{x}_{best}) + (1-\varphi) \cdot f(\vec{x}_{worst}), f(\vec{x}_i)\}, & i \in Z_2 \end{cases}$$

where φ denotes the feasibility proportion of the population.



CH for Semi-feasible Situation

Normalization of objective function value:

$$f_{nor}(\vec{x}_i) = \frac{f'(\vec{x}_i) - \min_{j \in Z_1 \cup Z_2} f'(\vec{x}_j)}{\max_{j \in Z_1 \cup Z_2} f'(\vec{x}_j) - \min_{j \in Z_1 \cup Z_2} f'(\vec{x}_j)}$$

Normalization of constraint violation value:

$$G_{nor}(\vec{x}_i) = \begin{cases} 0, & i \in Z_1 \\ \dfrac{G(\vec{x}_i) - \min_{j \in Z_2} G(\vec{x}_j)}{\max_{j \in Z_2} G(\vec{x}_j) - \min_{j \in Z_2} G(\vec{x}_j)}, & i \in Z_2 \end{cases}$$

The final fitness function:

$$f_{final}(\vec{x}_i) = f_{nor}(\vec{x}_i) + G_{nor}(\vec{x}_i)$$

IEEE PES Power & Energy Society® IEEE

Statistical Results

Case		Prob.		
		57 bus	118 bus	300 bus
ORPD	Best	24.6382	123.8418	400.0851
	Median	24.9043	127.9848	2.7223E+03
	Worst	26.9132	135.6955	6.9819E+04
	Mean	25.0414	128.3549	8.6696E+03
	Std	0.4296	2.4535	1.5256E+04
OARPD	Best	4.1703E+04	1.3624E+05	7.2261E+05
	Median	4.1738E+04	1.5741E+05	7.3910E+05
	Worst	4.1805E+04	1.7379E+05	7.5449E+05
	Mean	4.1740E+04	1.5423E+05	7.4011E+05
	Std	27.6884	1.2324E+04	6.2055E+03

Statistical Results

Scenario	Best	Median	Worst	Mean	Std
1	1.2972	1.2989	1.3114	1.2995	0.0025
2	1.2671	1.2697	1.2784	1.2703	0.0025
3	1.0134	1.0163	1.0510	1.0191	0.0078
4	1.2059	1.2094	1.2172	1.2096	0.0025
5	1.2747	1.2774	1.2868	1.2781	0.0030
6	1.5919	1.5948	1.6073	1.5958	0.0035
7	1.4285	1.4319	1.4367	1.4320	0.0021
8	1.4893	1.4918	1.4974	1.4923	0.0017
9	1.8989	1.9350	1.9814	1.9376	0.0196
10	2.0581	2.0969	2.1478	2.0958	0.0227
11	1.4546	1.5199	1.5889	1.5166	0.0365
12	1.6755	1.7146	1.7792	1.7169	0.0264

Statistical Results

Scenario	Best	Median	Worst	Mean	Std
13	1.7514	1.7803	1.8722	1.7871	0.0340
14	2.1170	2.1590	2.2262	2.1642	0.0293
15	2.5778	2.5951	2.6604	2.6015	0.0207
16	2.4962	2.5360	2.5846	2.5416	0.0213
17	2.7085	2.8034	2.9285	2.8052	0.0436
18	2.8121	2.9119	2.9723	2.9052	0.0396
19	2.8180	2.8943	2.9722	2.8952	0.0391
20	2.8133	2.9026	3.0116	2.8977	0.0455
21	2.7681	2.9109	2.9604	2.8995	0.0428
22	2.8070	2.9004	3.0651	2.9048	0.0513
23	2.8403	2.8965	2.9679	2.8942	0.0322
24	2.8225	2.8960	2.9454	2.8896	0.0362

Statistical Results

Scenario	Best	Median	Worst	Mean	Std
25	2.3017	2.3570	2.4277	2.3577	0.0296
26	2.2626	2.3062	2.3863	2.3077	0.0283
27	2.3165	2.3605	2.4119	2.3647	0.0240
28	2.5722	2.5975	2.6641	2.6016	0.0212
29	2.4116	2.4606	2.5259	2.4646	0.0282
30	2.4958	2.5304	2.5847	2.5324	0.0242
31	1.8317	1.8564	1.9773	1.8711	0.0341
32	1.6215	1.6607	1.7272	1.6678	0.0301
33	1.6186	1.6211	1.6233	1.6209	0.0012
34	1.6878	1.6895	1.6925	1.6897	0.0012
35	1.3567	1.3590	1.3621	1.3591	0.0011
36	1.3937	1.3964	1.3992	1.3966	0.0015

Statistical Results

Scenario	Best	Median	Worst	Mean	Std
37	0.9844	0.9864	1.0040	0.9872	0.0038
38	0.7595	0.7615	0.7653	0.7617	0.0012
39	0.7566	0.7578	0.7787	0.7590	0.0042
40	0.8385	0.8406	0.8506	0.8411	0.0021
41	0.9242	0.9284	0.9925	0.9369	0.0195
42	1.1524	1.1561	1.1895	1.1575	0.0066
43	1.2168	1.2199	1.2443	1.2212	0.0056
44	1.1955	1.1990	1.2337	1.2001	0.0065
45	2.0507	2.0938	2.1634	2.0952	0.0298
46	1.6319	1.6651	1.7236	1.6691	0.0286
47	1.5266	1.5875	1.6484	1.5900	0.0330
48	1.4523	1.5132	1.5638	1.5055	0.0320

Statistical Results

Scenario	Best	Median	Worst	Mean	Std
49	1.6189	1.6359	1.6514	1.6353	0.0075
50	1.4548	4008.2813	203543.5600	24280.2139	45671.1423
51	1.3922	10577.5460	295411.3600	90788.6183	109578.1524
52	129034.8900	341533.9000	1589444.3000	375966.7658	280330.0650
53	124029.3700	333694.9400	1276441.5000	448166.7548	299908.0942
54	127381.5200	338611.7000	1130953.5000	377030.7542	241218.1277
55	127408.0700	338177.2300	538760.8100	294390.3268	108906.4987
56	1.4638	3512.9198	78772.3410	9736.6190	22930.1212
57	1.1190	1.1706	1.2191	1.1688	0.0313
58	1.1260	1.1679	1.2213	1.1702	0.0269
59	1.1694	1.2351	1.2814	1.2302	0.0320
60	0.9691	1.0426	1.0820	1.0401	0.0282

Statistical Results

Scenario	Best	Median	Worst	Mean	Std
61	0.7572	0.7616	0.8036	0.7665	0.0109
62	0.7802	0.7869	0.8807	0.8017	0.0298
63	0.7706	0.7747	0.8357	0.7833	0.0180
64	0.9183	0.9215	0.9629	0.9277	0.0130
65	1.1181	1.1195	1.1252	1.1200	0.0016
66	1.3430	1.3448	1.3481	1.3448	0.0010
67	1.2371	1.2387	1.2491	1.2394	0.0022
68	0.8319	0.8340	0.8370	0.8342	0.0013
69	0.8643	0.8661	0.8784	0.8673	0.0035
70	1.0744	1.0755	1.0786	1.0756	0.0010
71	1.1324	1.1341	1.1416	1.1346	0.0022
72	1.1308	1.1317	1.1373	1.1326	0.0018

Statistical Results

Scenario	Best	Median	Worst	Mean	Std
73	1.4209	1.4244	1.4299	1.4251	0.0028
74	1.7814	1.7842	1.7949	1.7852	0.0034
75	2.0126	2.0251	821.4249	29.6791	147.0538
76	66330.4480	422096.5500	1199290.6000	389506.3503	227902.9656
77	71976.5040	206531.0800	747178.6200	328088.3163	198614.8268
78	64177.1200	204502.9900	758912.3800	318576.4135	194083.7729
79	64560.8700	204951.6600	748231.0100	293711.5866	169600.3436
80	65566.9490	208404.6100	754941.3100	334114.7159	184051.6378
81	2.4623	2.4771	4104.6736	134.8449	736.7690
82	1.5214	1.5254	1.5272	1.5250	0.0013
83	1.5462	1.5493	1.5530	1.5493	0.0019
84	1.4072	1.4098	1.4122	1.4097	0.0015

Statistical Results

Scenario	Best	Median	Worst	Mean	Std
85	1.2800	1.2825	1.2867	1.2823	0.0013
86	1.3209	1.3230	1.3253	1.3231	0.0011
87	1.5615	1.5638	1.5658	1.5638	0.0013
88	1.5551	1.5580	1.5609	1.5578	0.0015
89	1.4737	1.4774	1.4883	1.4784	0.0033
90	2.0095	2.0162	2.0288	2.0174	0.0056
91	2.1827	2.1965	2.2089	2.1951	0.0075
92	1.6195	1.6223	1.6326	1.6235	0.0033
93	2.0576	2.0644	2.0781	2.0655	0.0056
94	2.4949	2.5118	2.5179	2.5116	0.0038
95	1.7847	1.7928	1.8044	1.7933	0.0044
96	1.5013	1.5032	1.5075	1.5037	0.0017

THANK YOU