Balancing handover areas among neighbouring cells in 3G networks

Evolutionary Algorithms

Multiobjective optimization and design

Lucas Benedičič

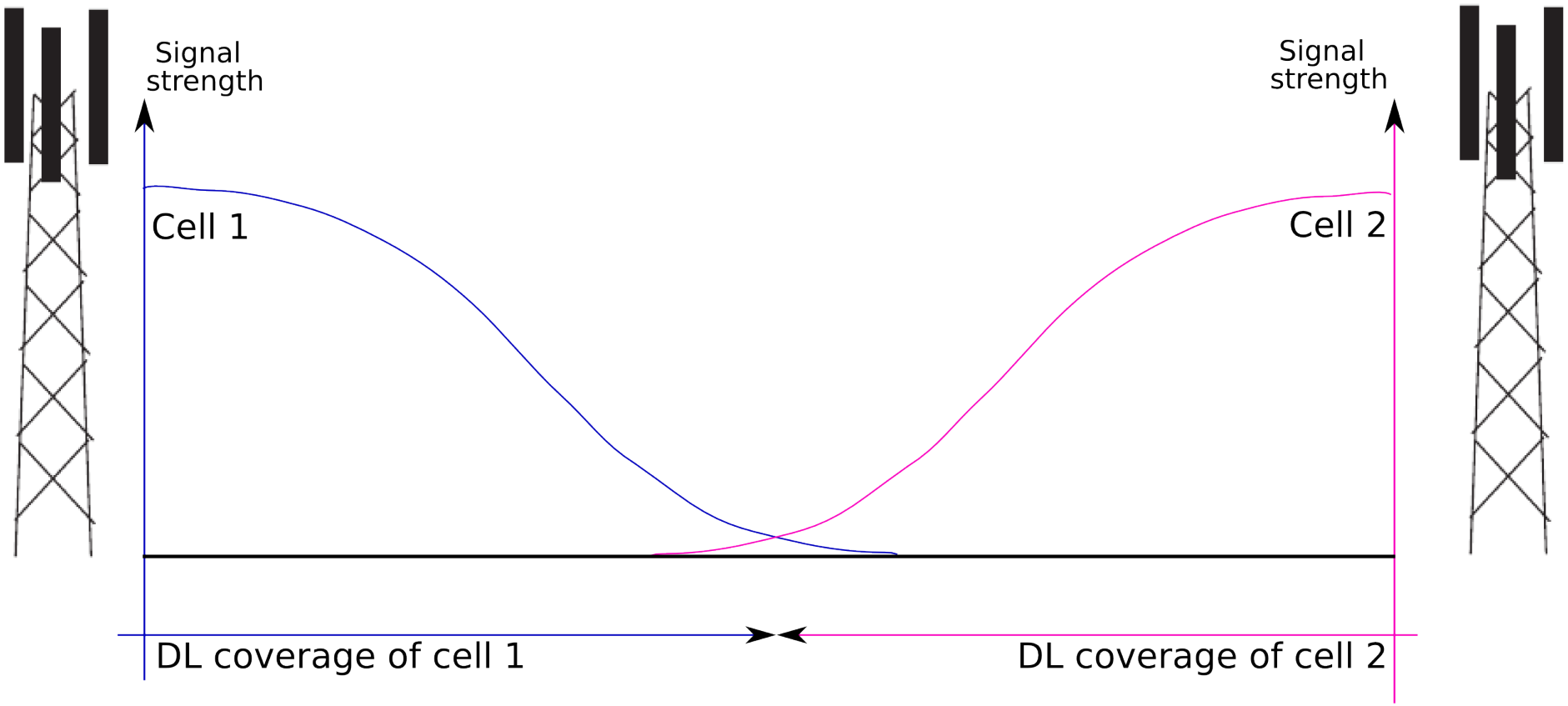
Jožef Stefan International Postgraduate School

May 2010

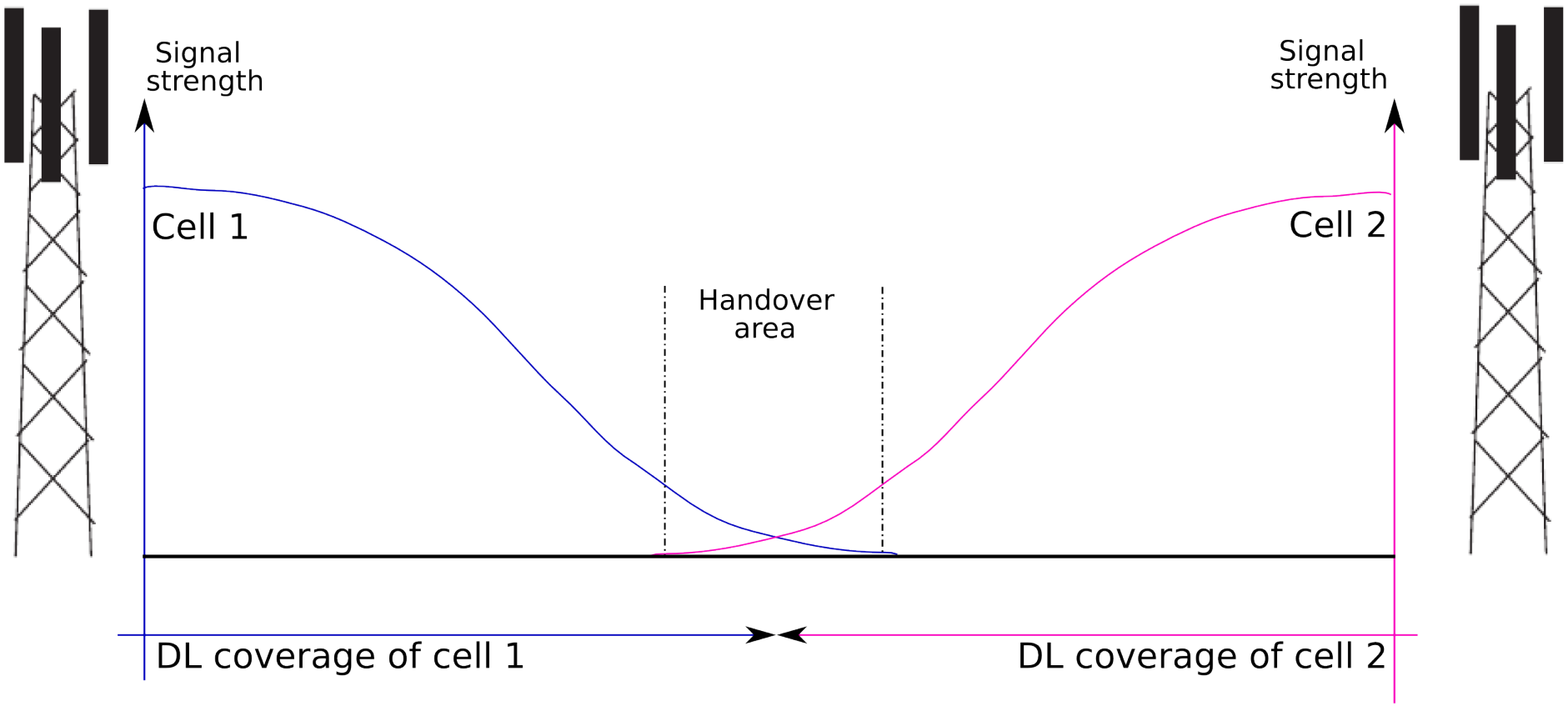
Outline

Outline

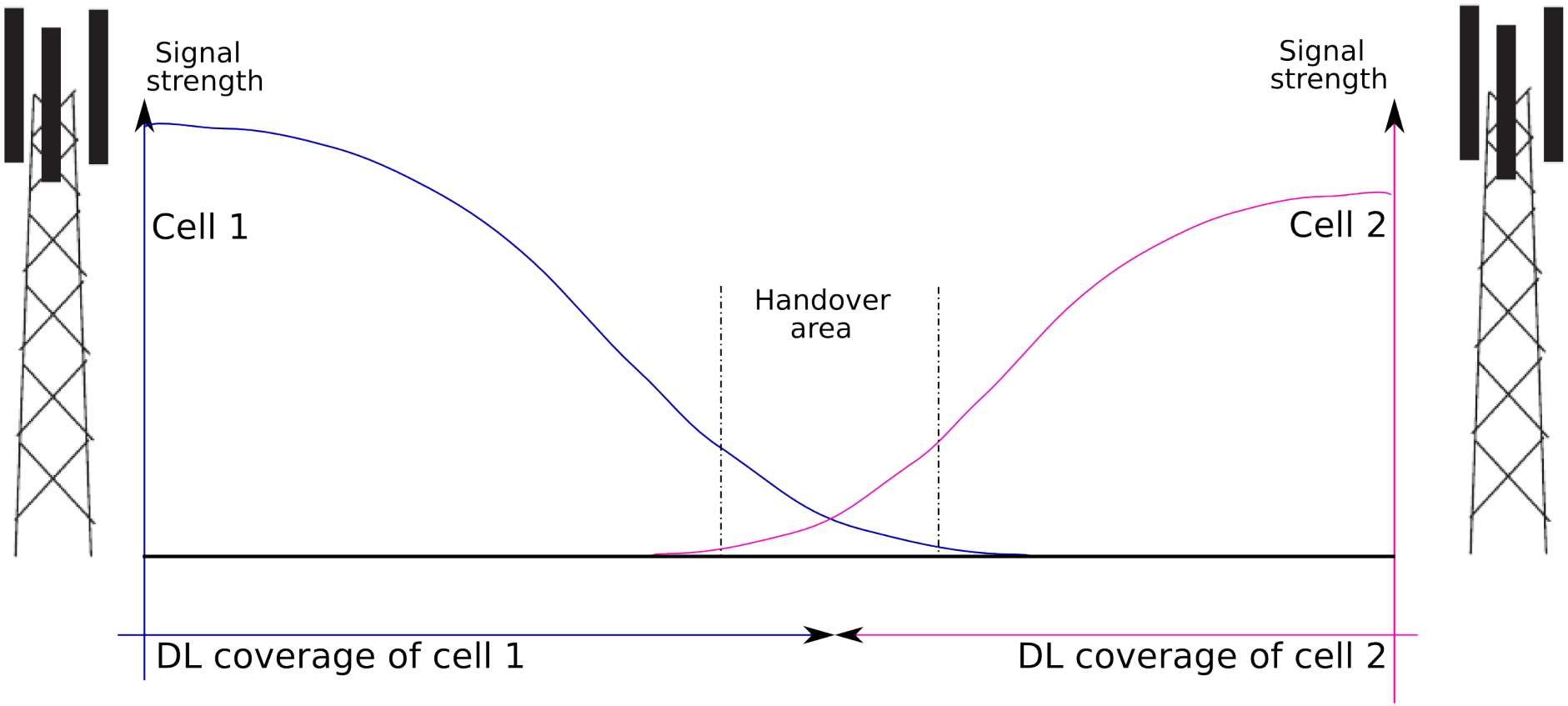
Neighbour cells



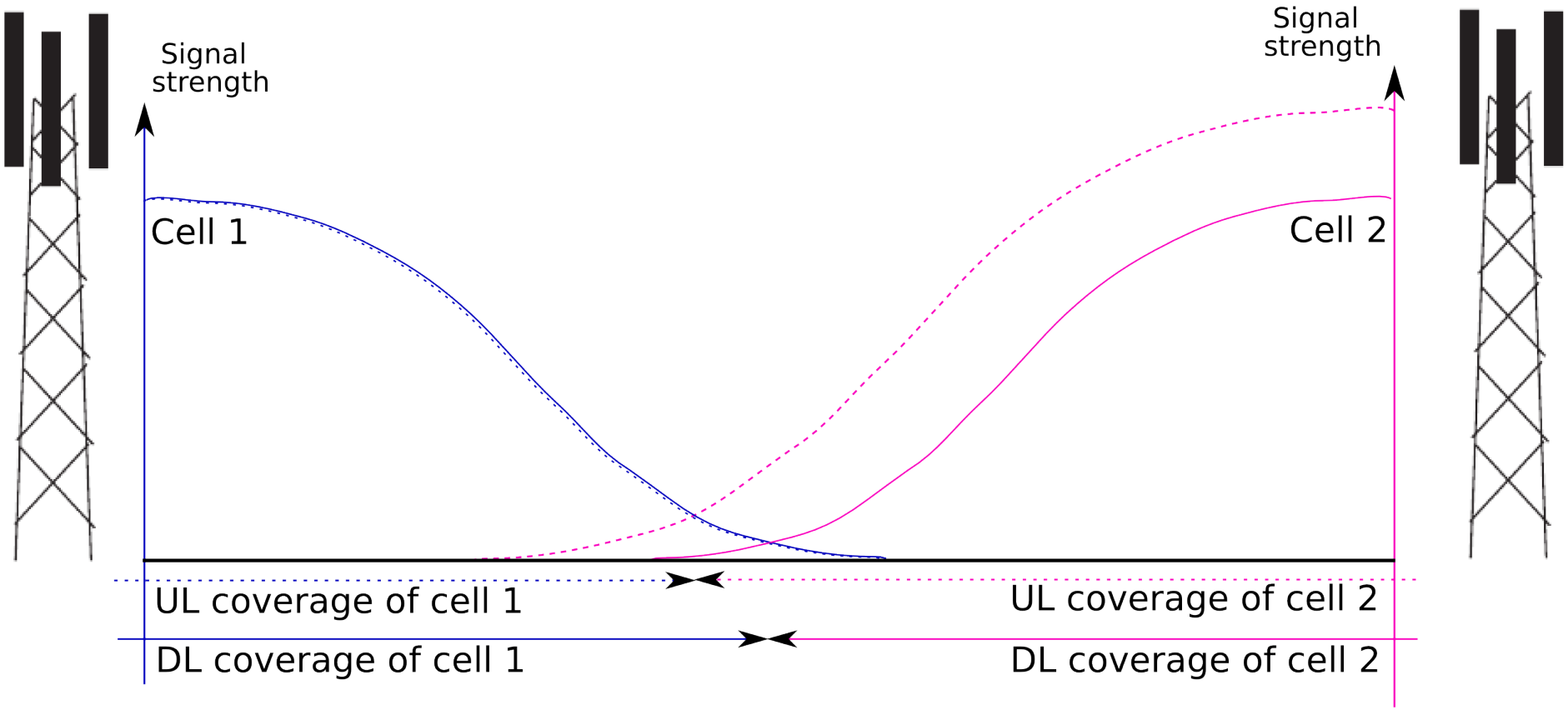
Handover area



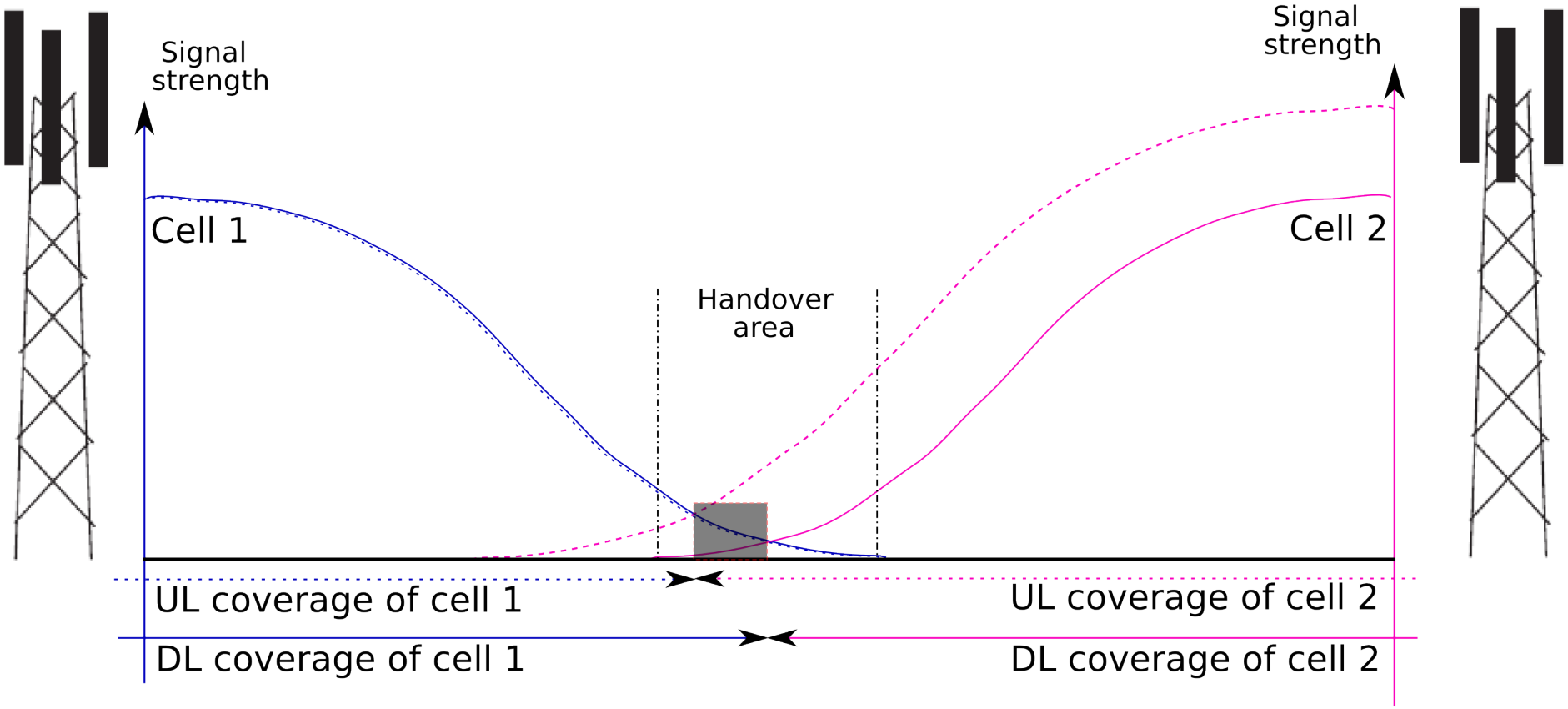
Balancing handover area



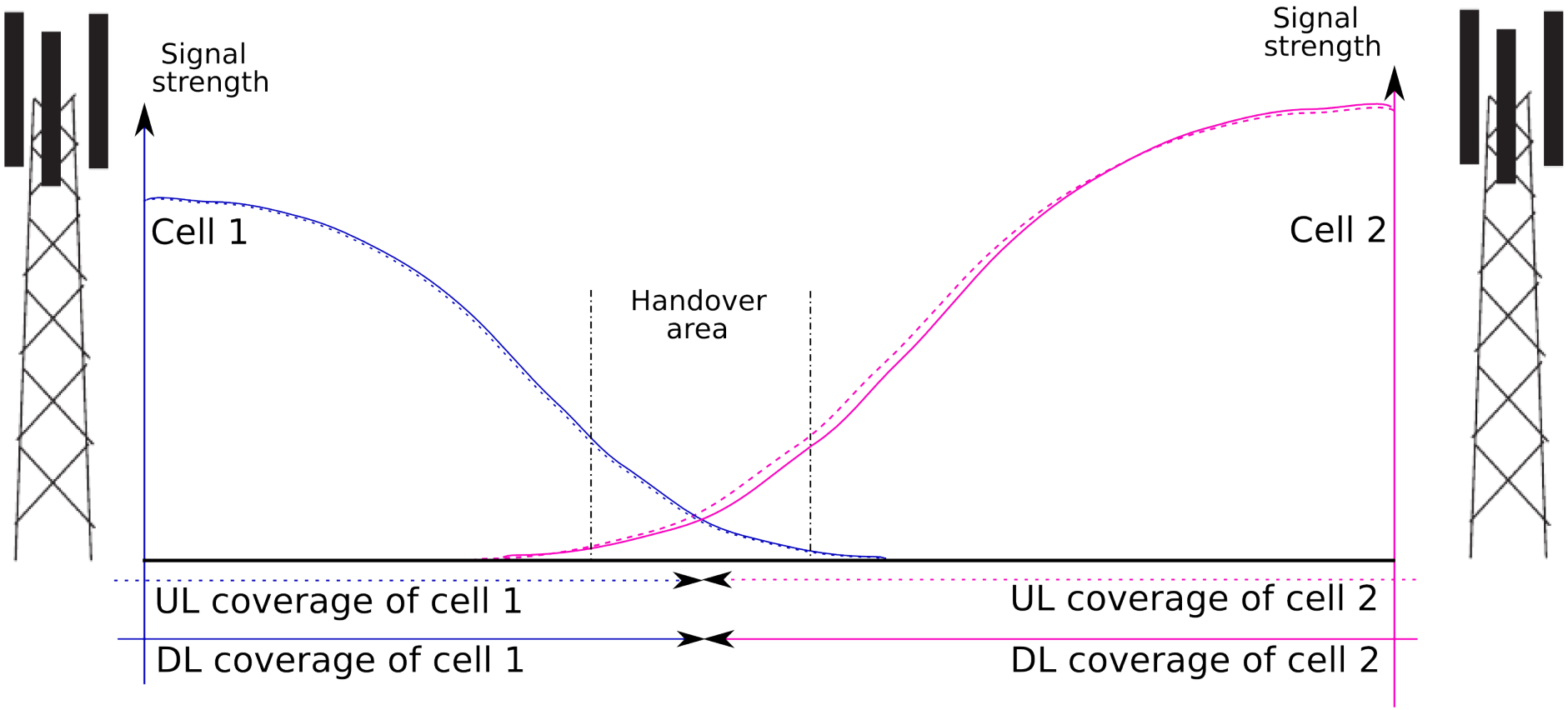
Balancing UL-DL handover areas



Balancing UL-DL handover areas



Balancing UL-DL handover areas



Outline

Adjacency Matrix

	CELL 1	CELL 2	CELL 3	CELL 4	CELL 5
CELL 1					
CELL 2					
CELL 3					
CELL 4					
CELL 5					

Adjacency Matrix

	CELL 1	CELL 2	CELL 3	CELL 4	CELL 5
CELL 1	0	0	1	0	1
CELL 2	0	0	0	1	0
CELL 3	1	0	0	0	1
CELL 4	0	1	0	0	0
CELL 5	1	0	1	0	0

Adjacency Matrix - simmetry

	CELL 1	CELL 2	CELL 3	CELL 4	CELL 5
CELL 1	0	0	1	0	1
CELL 2	0	0	0	1	0
CELL 3	1	0	0	0	1
CELL 4	0	1	0	0	0
CELL 5	1	0	1	0	0

Outline

Problem elements

- **Candidate solutions:** DL power setting for every cell.
- **Constraints:** solutions within $[-2dB, +2dB]$ of current setting.
- Static problem.

Problem elements

- **Candidate solutions:** DL power setting for every cell.
- **Constraints:** solutions within $[-2dB, +2dB]$ of current setting.
- Static problem.

Problem elements

- **Candidate solutions:** DL power setting for every cell.
- **Constraints:** solutions within $[-2dB, +2dB]$ of current setting.
- Static problem.

Objective function

Minimize ...

... the difference between UL and DL areas among neighbour cells.

Objective function

Objective function

$$\min \left| \sum_{i=1}^{|V|} \sum_{j=i}^{|V|} a_{ij} \left((UL(c_i) - UL(c_j)) - (DL(c_i) - DL(c_j)) \right) \right|$$

- c_i i-th cell of the network,
- $DL(c_i)$ downlink power of the i-th cell,
- $UL(c_i)$ uplink power of the i-the cell.

Objective function

Objective function

$$\min \left| \sum_{i=1}^{|V|} \sum_{j=i}^{|V|} a_{ij} \left((UL(c_i) - UL(c_j)) - (DL(c_i) - DL(c_j)) \right) \right|$$

- c_i i-th cell of the network,
- $DL(c_i)$ downlink power of the i-th cell,
- $UL(c_i)$ uplink power of the i-the cell.

Objective function

Objective function

$$\min \left| \sum_{i=1}^{|V|} \sum_{j=i}^{|V|} a_{ij} \left((UL(c_i) - UL(c_j)) - (DL(c_i) - DL(c_j)) \right) \right|$$

- c_i i-th cell of the network,
- $DL(c_i)$ downlink power of the i-th cell,
- $UL(c_i)$ uplink power of the i-the cell.

Objective function

Objective function

$$\min \left| \sum_{i=1}^{|V|} \sum_{j=i}^{|V|} a_{ij} \left((UL(c_i) - UL(c_j)) - (DL(c_i) - DL(c_j)) \right) \right|$$

- c_i i-th cell of the network,
- $DL(c_i)$ downlink power of the i-th cell,
- $UL(c_i)$ uplink power of the i-the cell.

Objective function

Objective function

$$\min \left| \sum_{i=1}^{|V|} \sum_{j=i}^{|V|} a_{ij} \left((UL(c_i) - UL(c_j)) - (DL(c_i) - DL(c_j)) \right) \right|$$

- a_{ij} adjacency matrix coefficient at (i, j),
- $|V|$ total number of cells in the network.

Objective function

Objective function

$$\min \left| \sum_{i=1}^{|V|} \sum_{j=i}^{|V|} a_{ij} \left((UL(c_i) - UL(c_j)) - (DL(c_i) - DL(c_j)) \right) \right|$$

- a_{ij} adjacency matrix coefficient at (i, j),
- $|V|$ total number of cells in the network.

Objective function

Objective function

$$\min \left| \sum_{i=1}^{|V|} \sum_{j=i}^{|V|} a_{ij} \left((UL(c_i) - UL(c_j)) - (DL(c_i) - DL(c_j)) \right) \right|$$

- a_{ij} adjacency matrix coefficient at (i, j),
- $|V|$ total number of cells in the network.

Simulated annealing

- Discrete search space
  - changes of $0.1dBm$.
- Key parameters
  - initial temperature,
  - number of iterations.

Simulated annealing

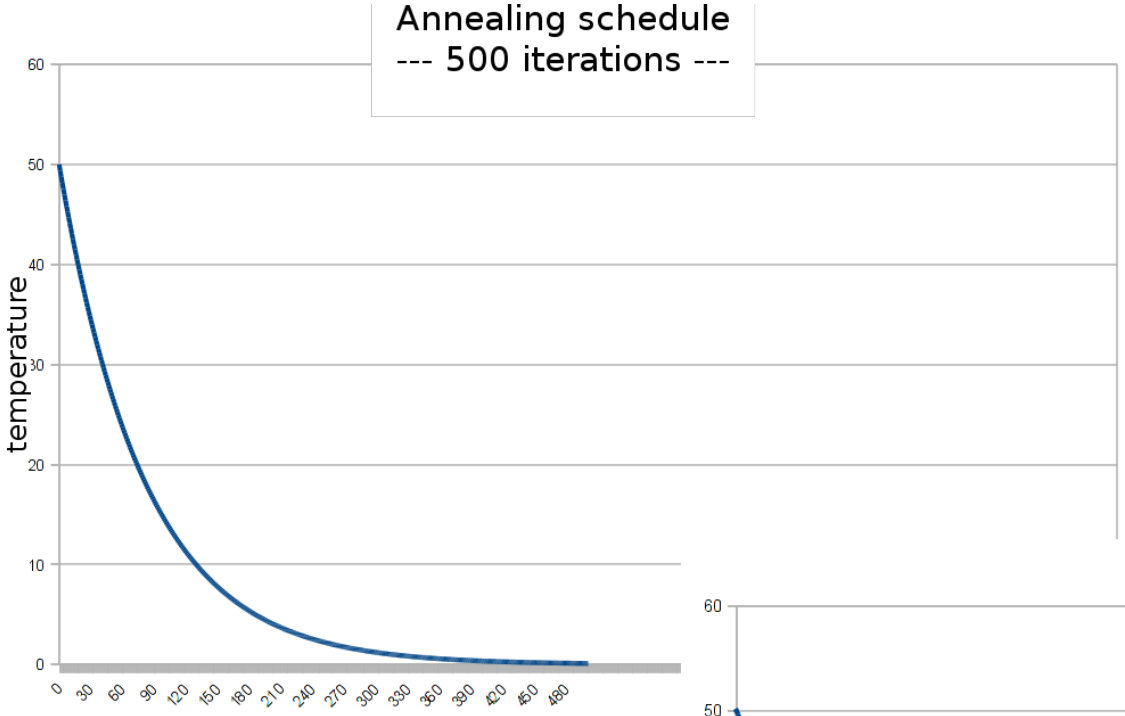
- Discrete search space
  - changes of $0.1dBm$.
- Key parameters
  - initial temperature,
  - number of iterations.

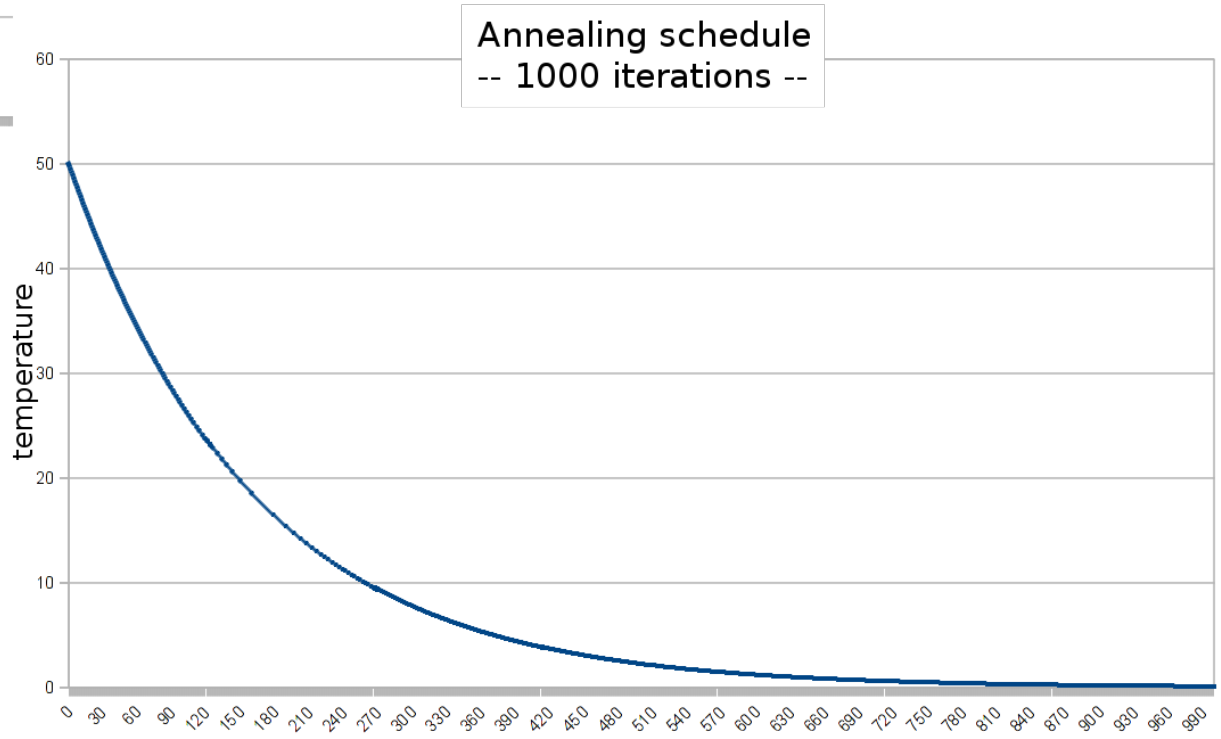
Simulated annealing

- Discrete search space
  - changes of $0.1dBm$.
- Key parameters
  - initial temperature,
  - number of iterations.

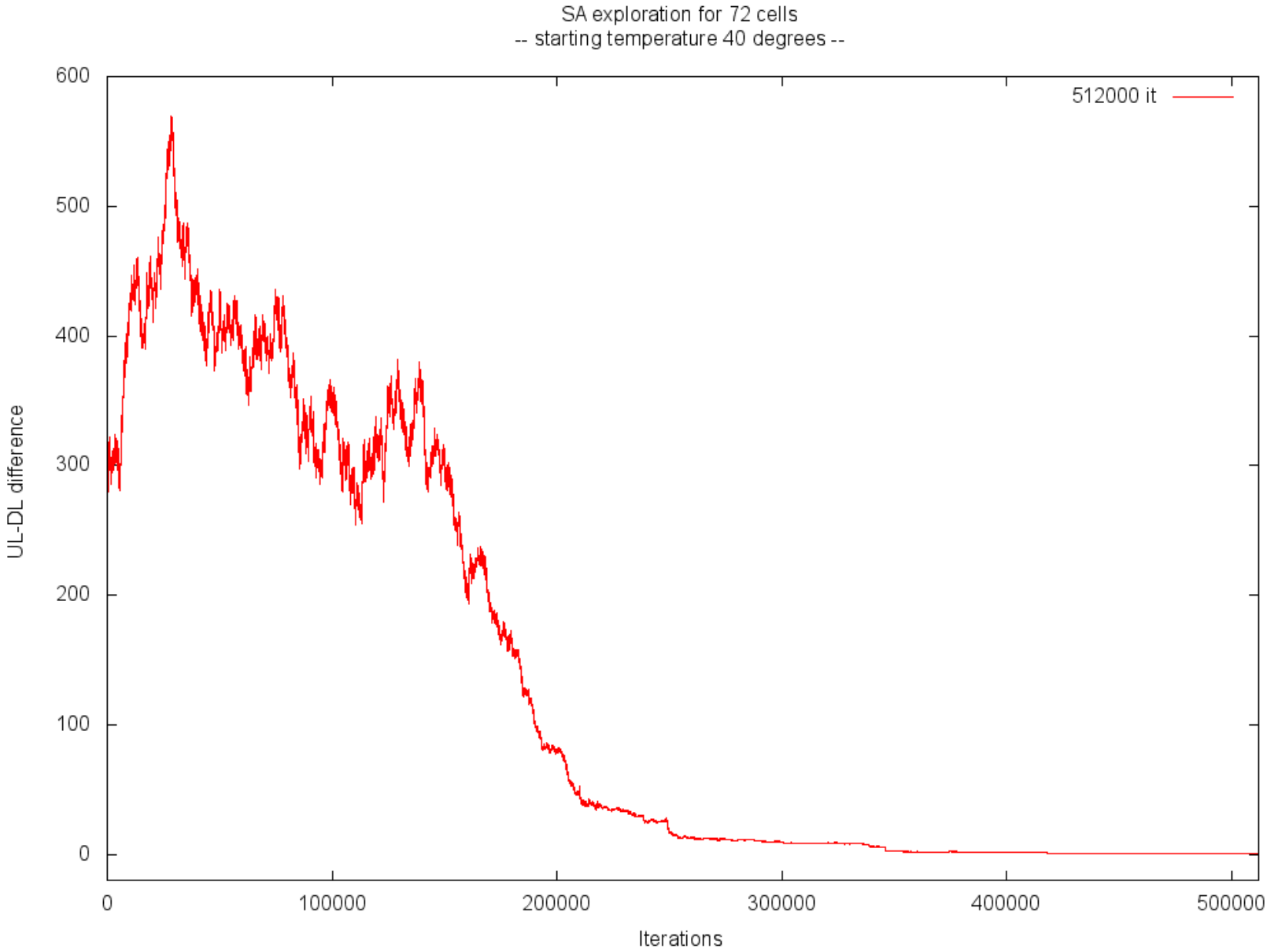
Simulated annealing

- Discrete search space
  - changes of $0.1 dBm$.
- Key parameters
  - initial temperature,
  - number of iterations.


Annealing schedule
--- 500 iterations ---
temperature
60
50
40
30
20
10
0
0 30 60 90 120 150 180 210 240 270 300 330 360 390 420 450 480


Annealing schedule
-- 1000 iterations --
temperature
60
50
40
30
20
10
0
0 30 60 90 120 150 180 210 240 270 300 330 360 390 420 450 480 510 540 570 600 630 660 690 720 750 780 810 840 870 900 930 960 990

SA: Results


SA exploration for 72 cells
-- starting temperature 40 degrees --
512000 it
UL-DL difference
Iterations
0
100
200
300
400
500
600
0
100000
200000
300000
400000
500000

Objective functions

Minimize ...

... the difference between UL and DL areas among neighbour cells.

Minimize ...

... the total DL power used.

Objective functions

1 - Objective function

$$\min \left| \sum_{i=1}^{|V|} \sum_{j=i}^{|V|} a_{ij} \left((UL(c_i) - UL(c_j)) - (DL(c_i) - DL(c_j)) \right) \right|$$

2 - Objective function

$$\min \sum_{i=1}^{|V|} DL(c_i)$$

Objective functions

1 - Objective function

$$\min \left| \sum_{i=1}^{|V|} \sum_{j=i}^{|V|} a_{ij} \left((UL(c_i) - UL(c_j)) - (DL(c_i) - DL(c_j)) \right) \right|$$

2 - Objective function

$$\min \sum_{i=1}^{|V|} DL(c_i)$$

DEMO

- Continous search space.
- Key parameters
  - population size,
  - number of generations.

DEMO

- Continous search space.
- Key parameters
  - population size,
  - number of generations.

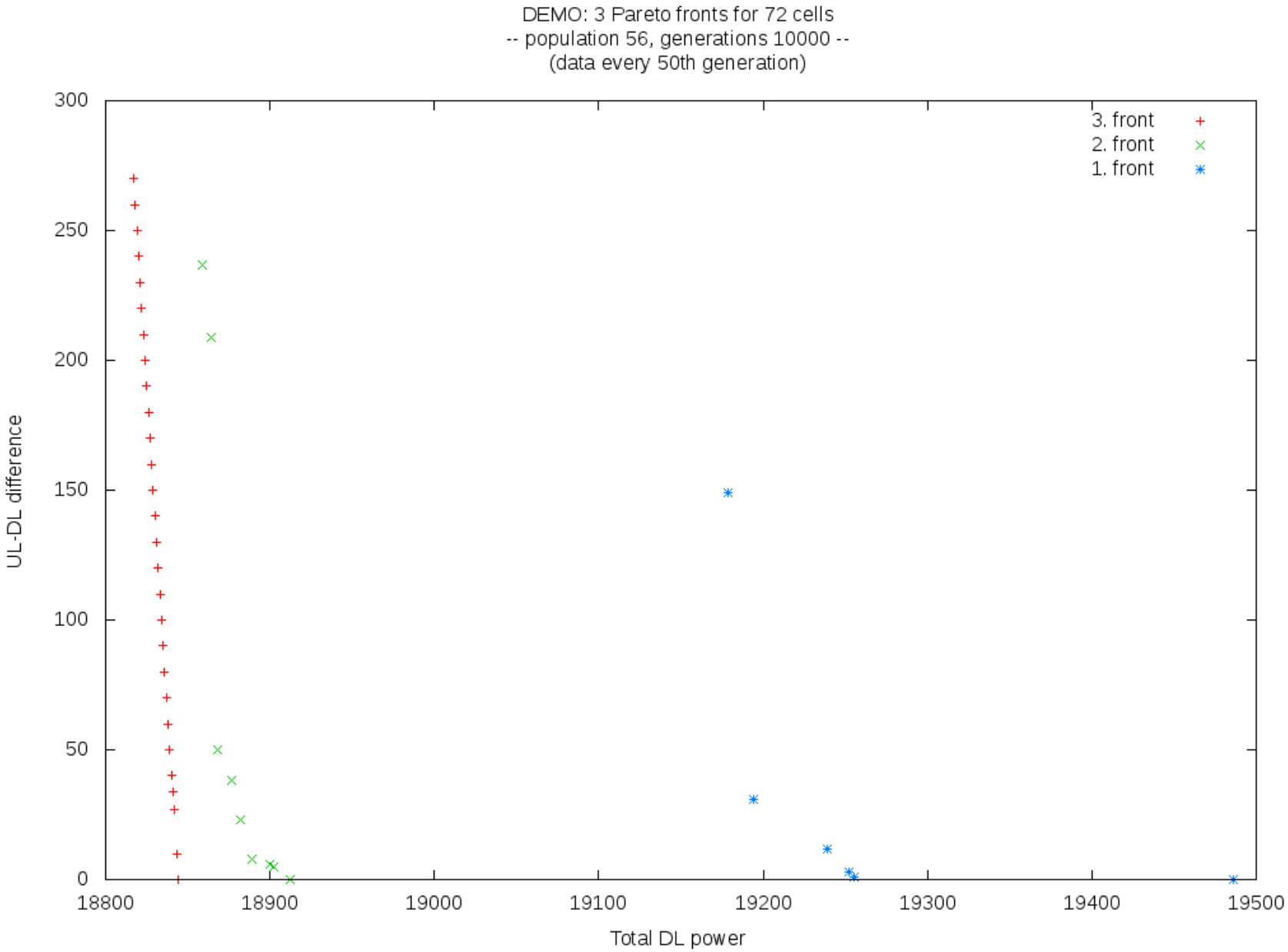
DEMO

- Continous search space.
- Key parameters
  - population size,
  - number of generations.

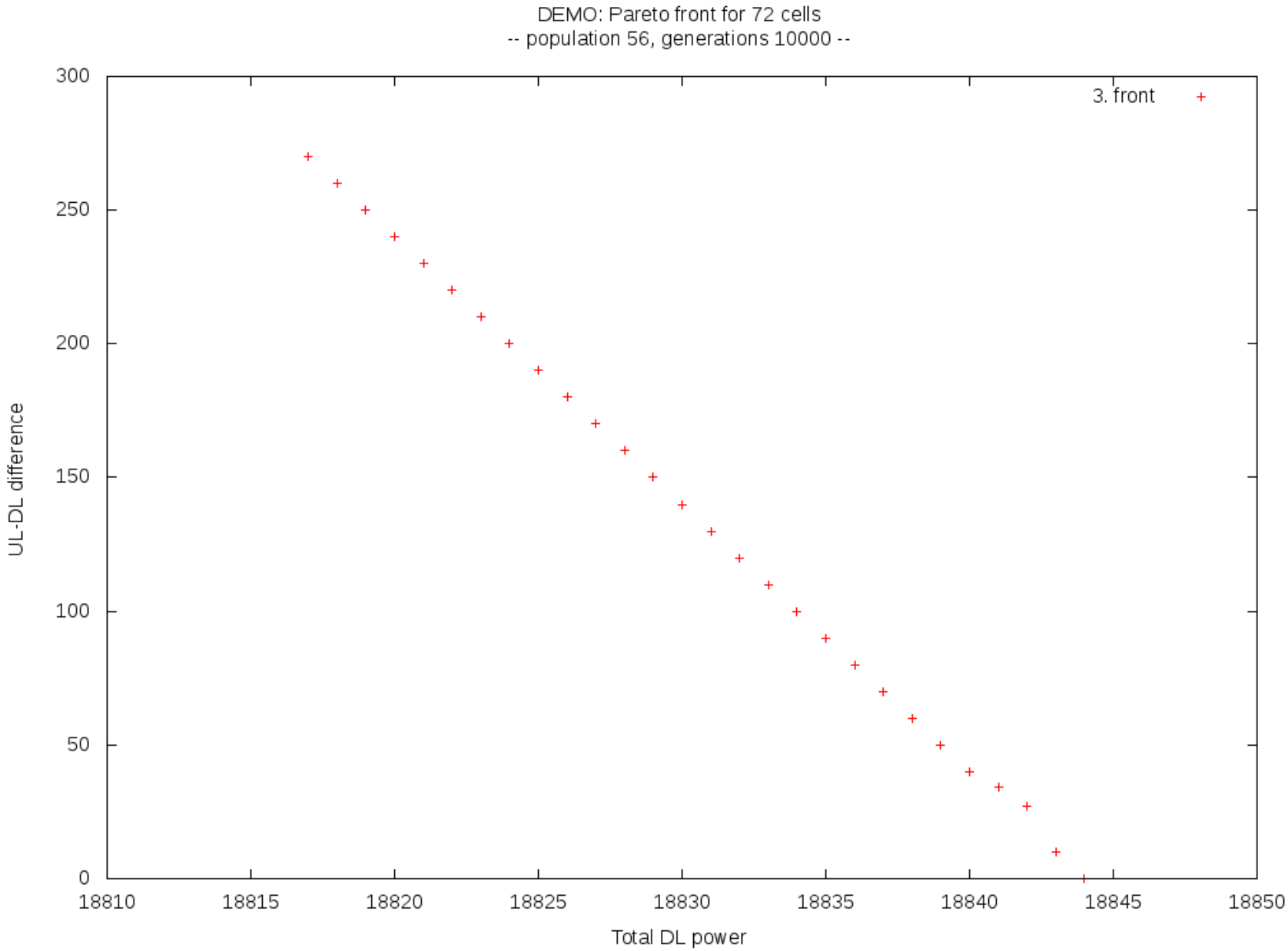
DEMO

- Continous search space.
- Key parameters
  - population size,
  - number of generations.

DEMO: Results



DEMO: Results



Concluding remarks

- Both algorithms solved the balancing problem to optimality.
- SA was easier to set-up (implementation + parameter configuration).
- SA took longer running times for comparable good results.

Concluding remarks

- Both algorithms solved the balancing problem to optimality.
- SA was easier to set-up (implementation + parameter configuration).
- SA took longer running times for comparable good results.

Concluding remarks

- Both algorithms solved the balancing problem to optimality.
- SA was easier to set-up (implementation + parameter configuration).
- SA took longer running times for comparable good results.

Concluding remarks

- DEMO analysis included 2 parameters only (lots of other parameters left as default).
- DEMO was much faster (minutes $\longleftrightarrow$ hours).
- DEMO also lowered the total power used (2^{nd} objective).
- DEMO couldn't find good solutions in a discretized search space.

Concluding remarks

- DEMO analysis included 2 parameters only (lots of other parameters left as default).
- DEMO was much faster (minutes $\longleftrightarrow$ hours).
- DEMO also lowered the total power used (2^{nd} objective).
- DEMO couldn't find good solutions in a discretized search space.

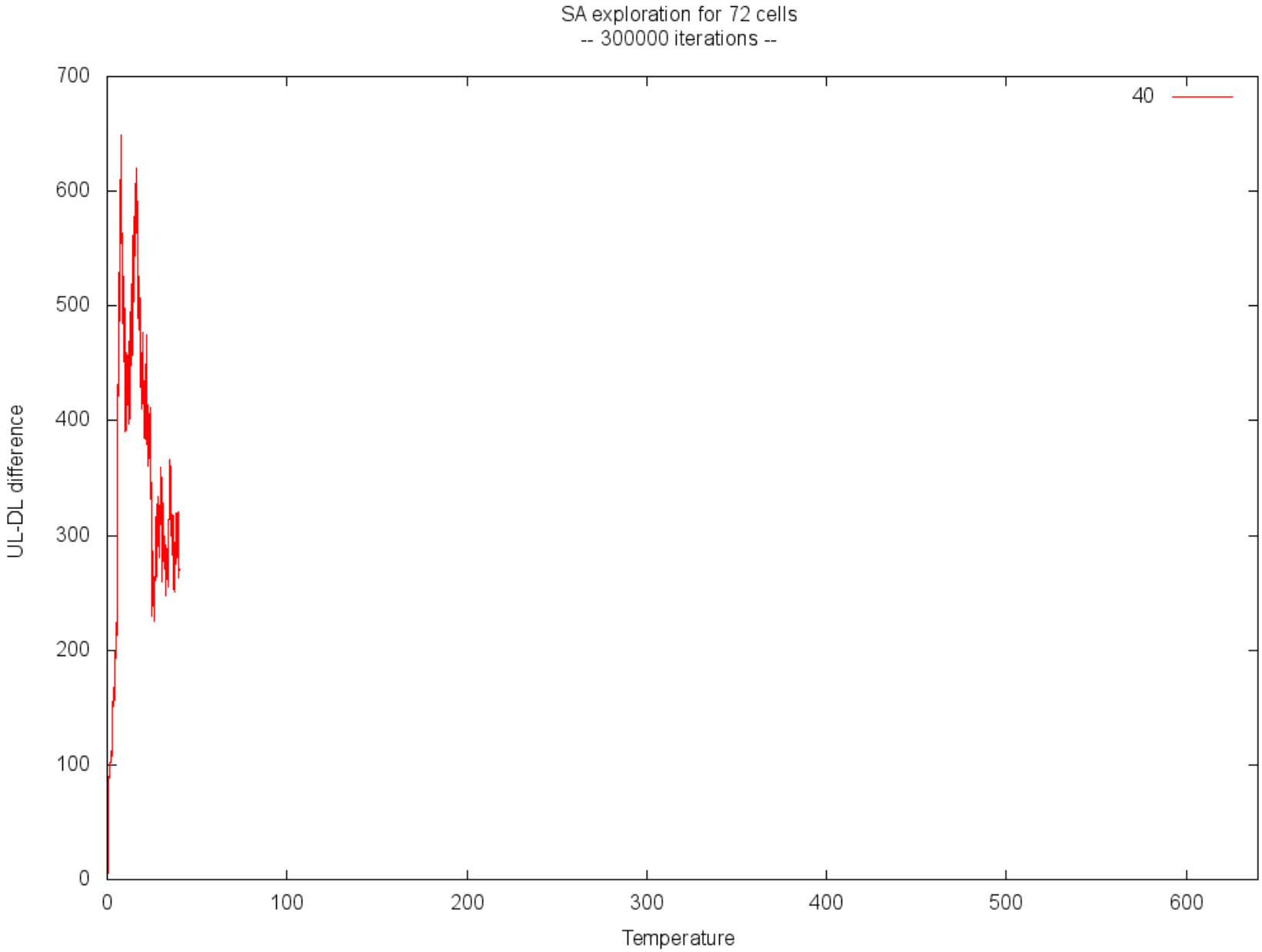
Concluding remarks

- DEMO analysis included 2 parameters only (lots of other parameters left as default).
- DEMO was much faster (minutes $\longleftrightarrow$ hours).
- DEMO also lowered the total power used (2^{nd} objective).
- DEMO couldn't find good solutions in a discretized search space.

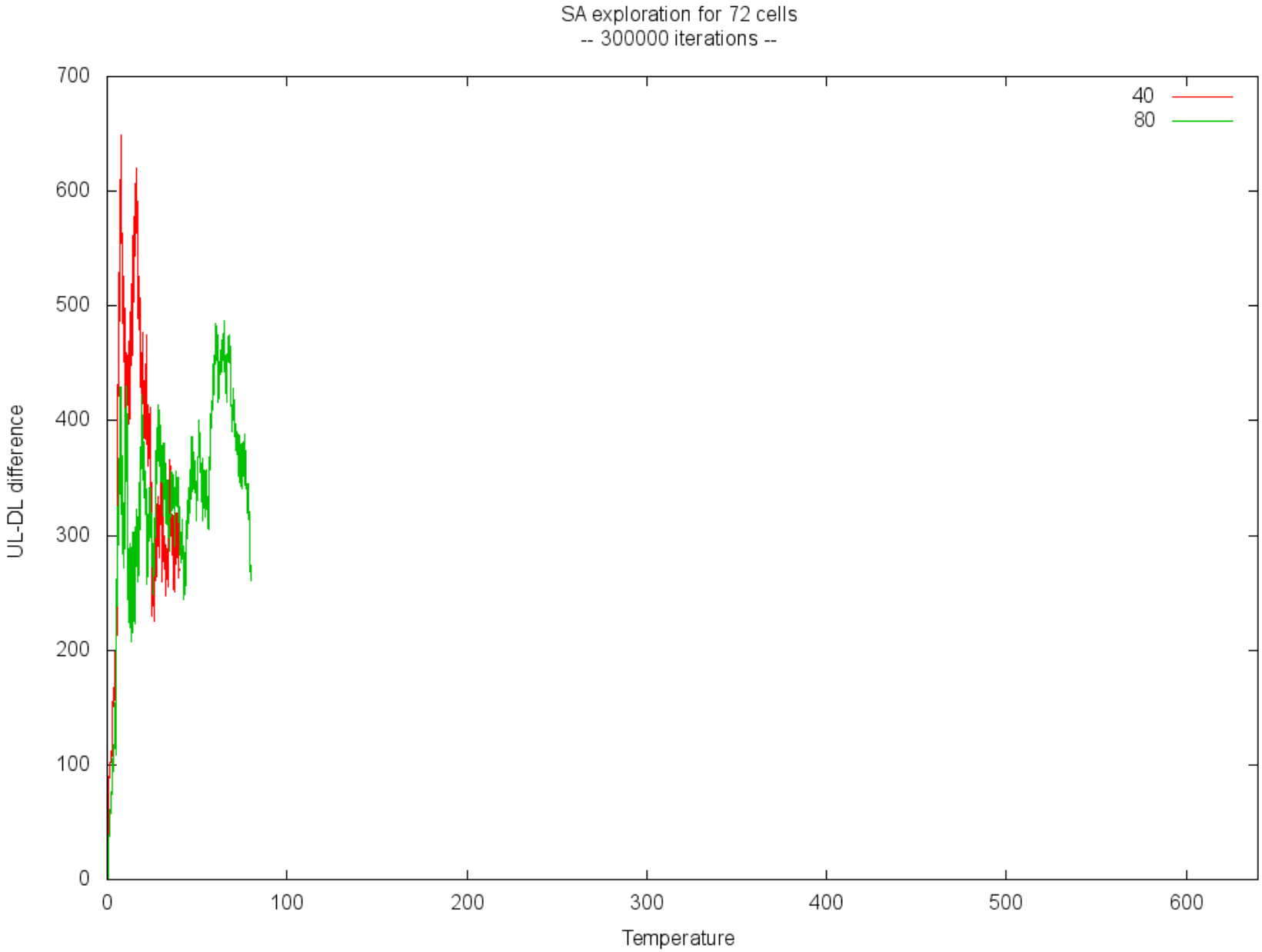
Concluding remarks

- DEMO analysis included 2 parameters only (lots of other parameters left as default).
- DEMO was much faster (minutes $\longleftrightarrow$ hours).
- DEMO also lowered the total power used (2^{nd} objective).
- DEMO couldn't find good solutions in a discretized search space.

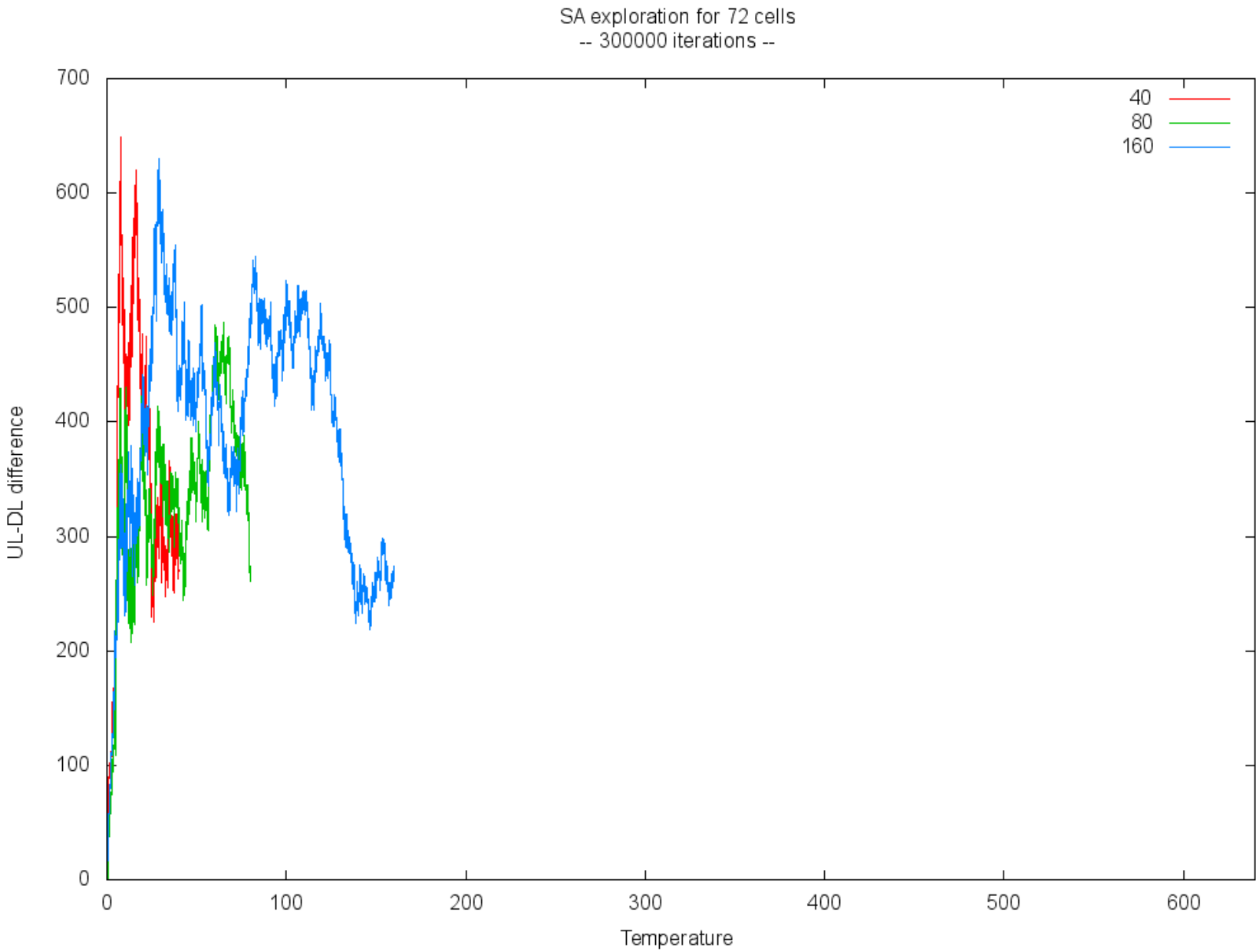
SA: Starting temperature



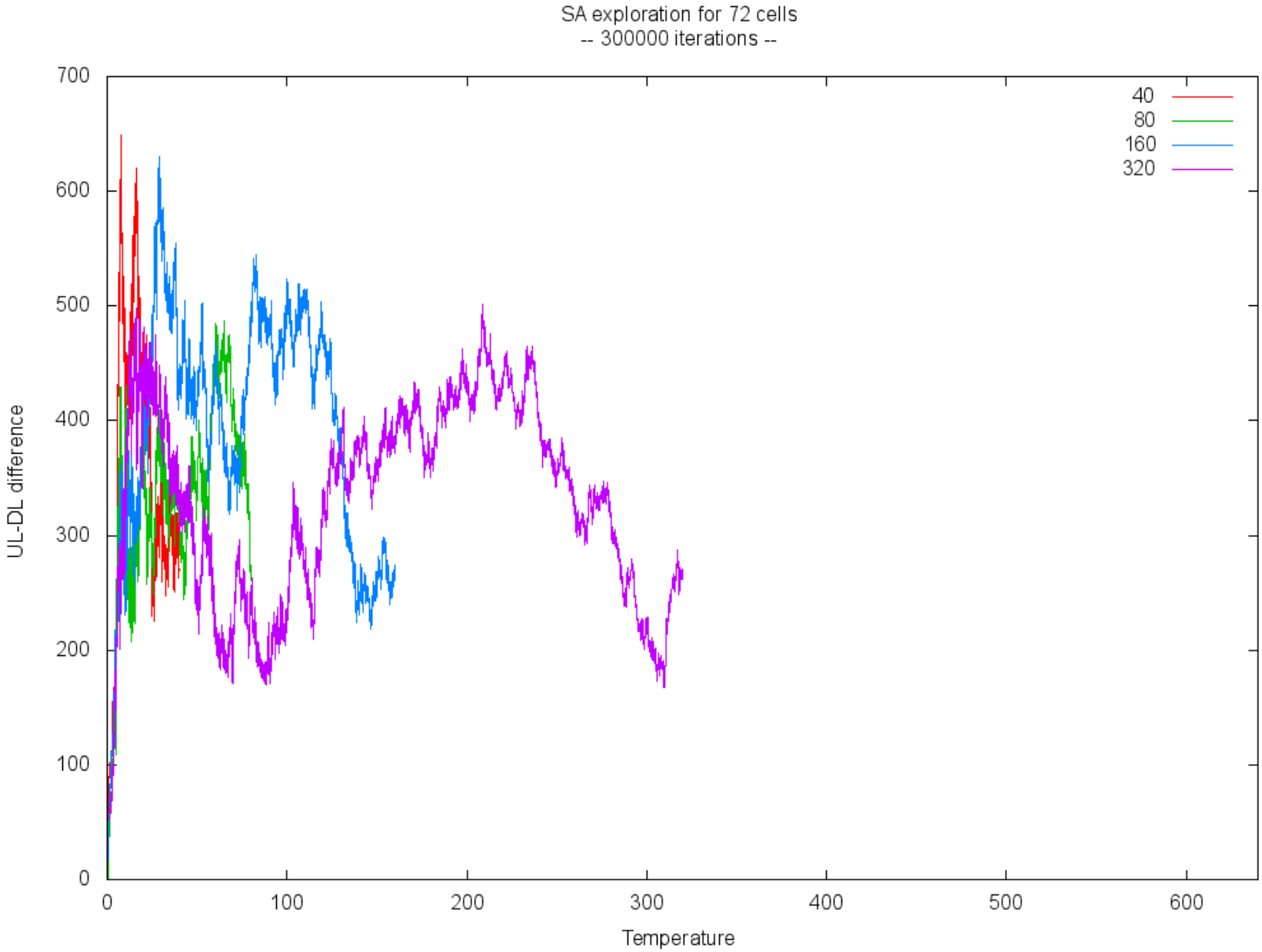
SA: Starting temperature



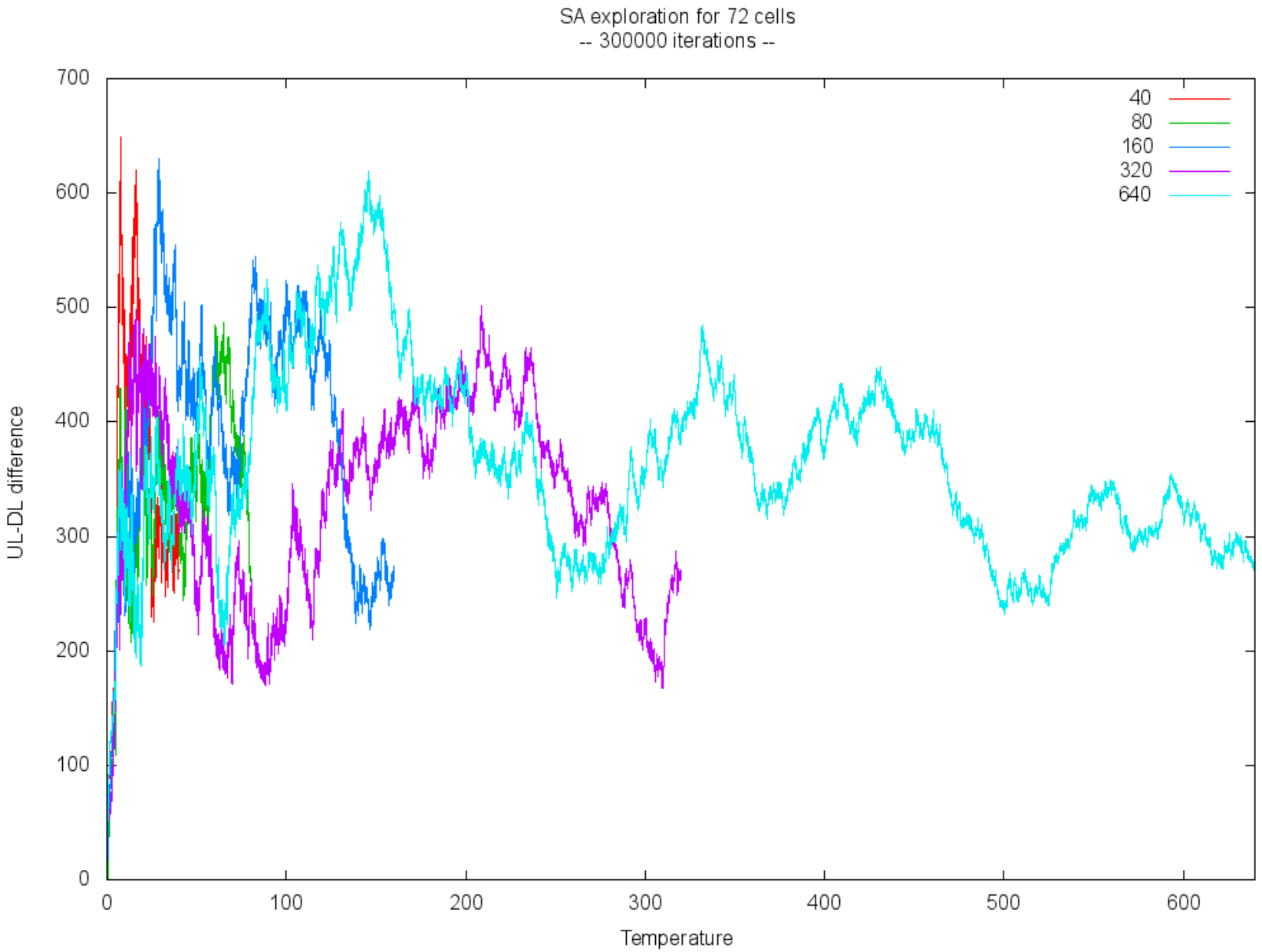
SA: Starting temperature



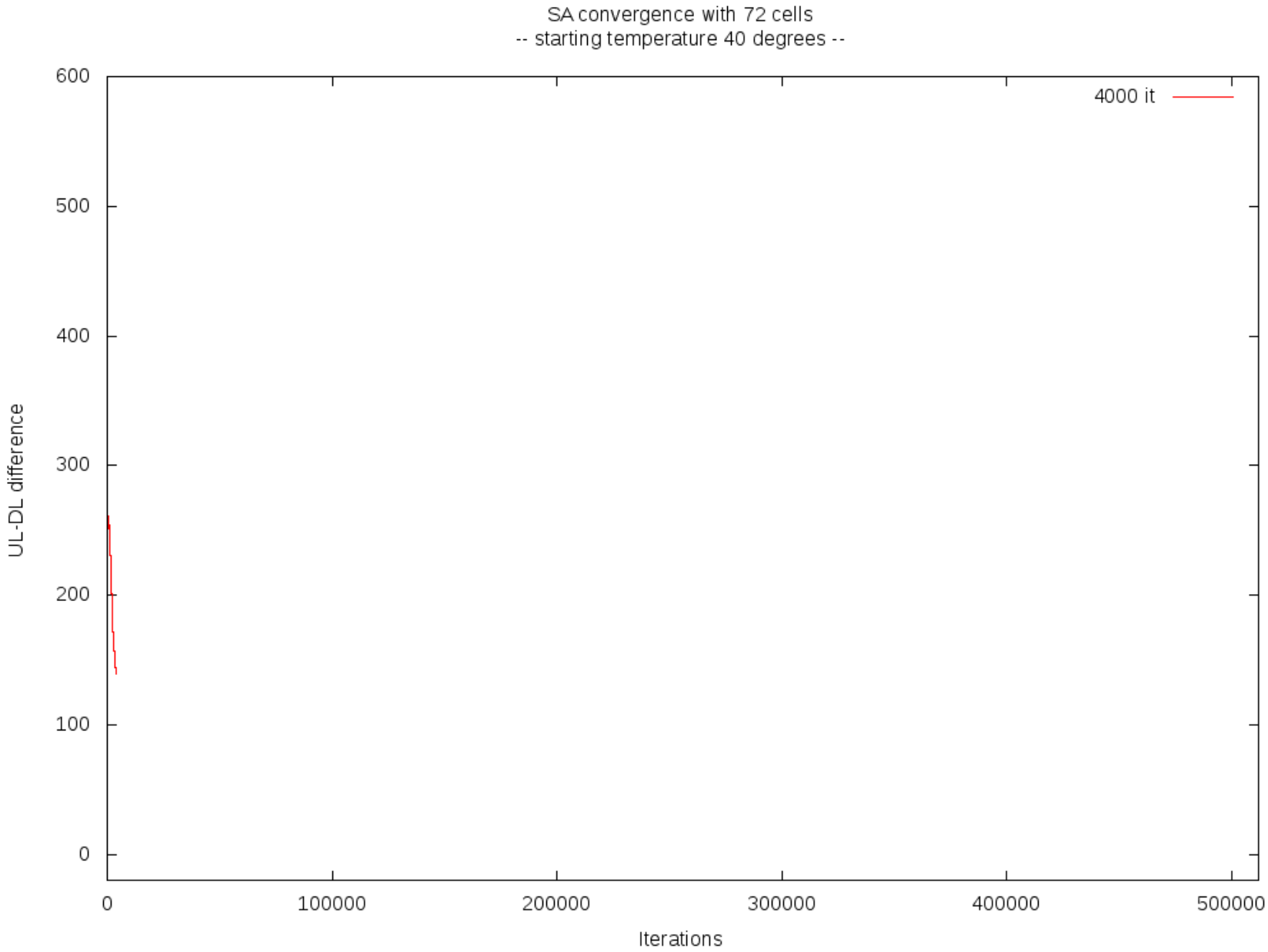
SA: Starting temperature



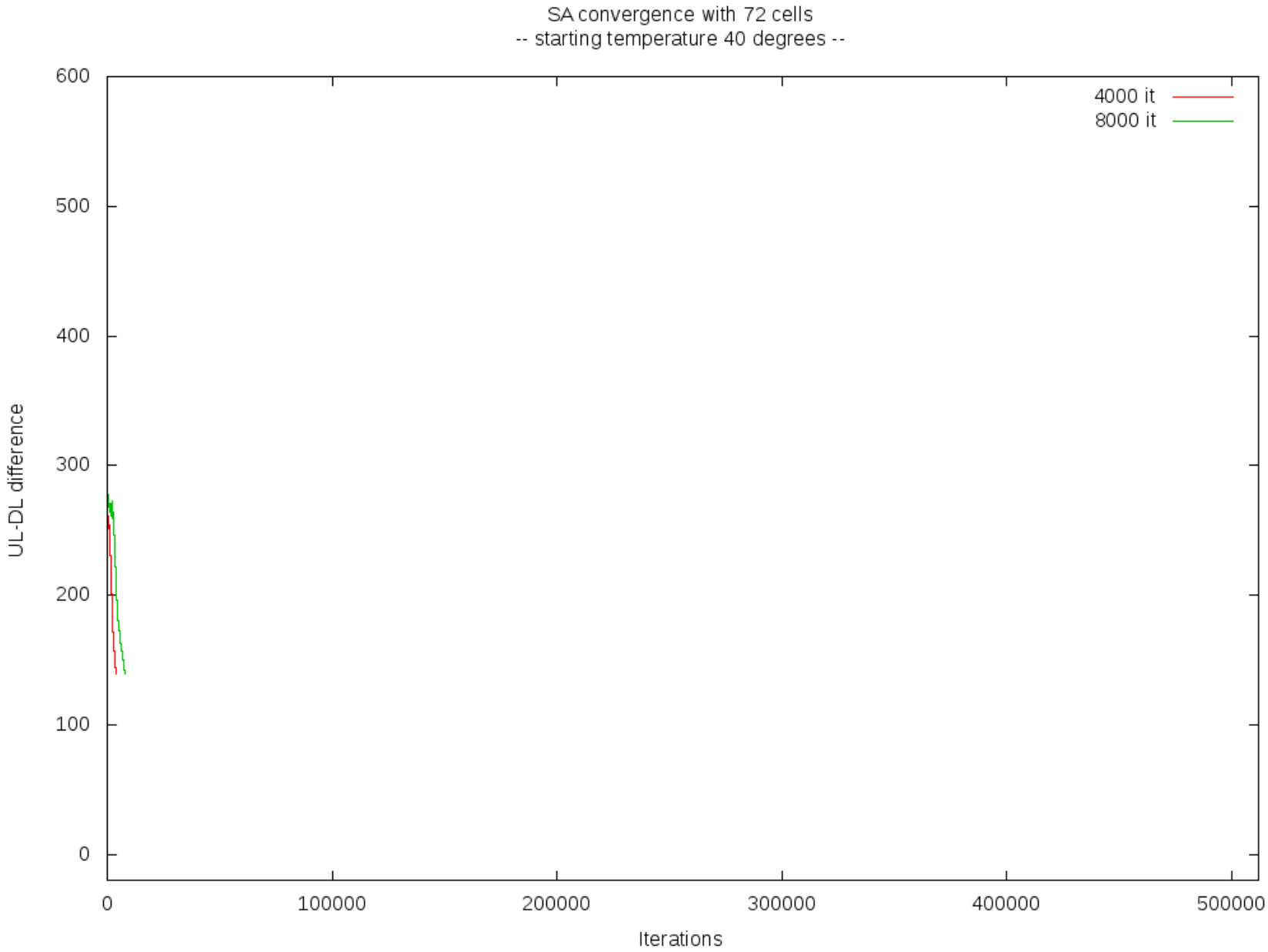
SA: Starting temperature



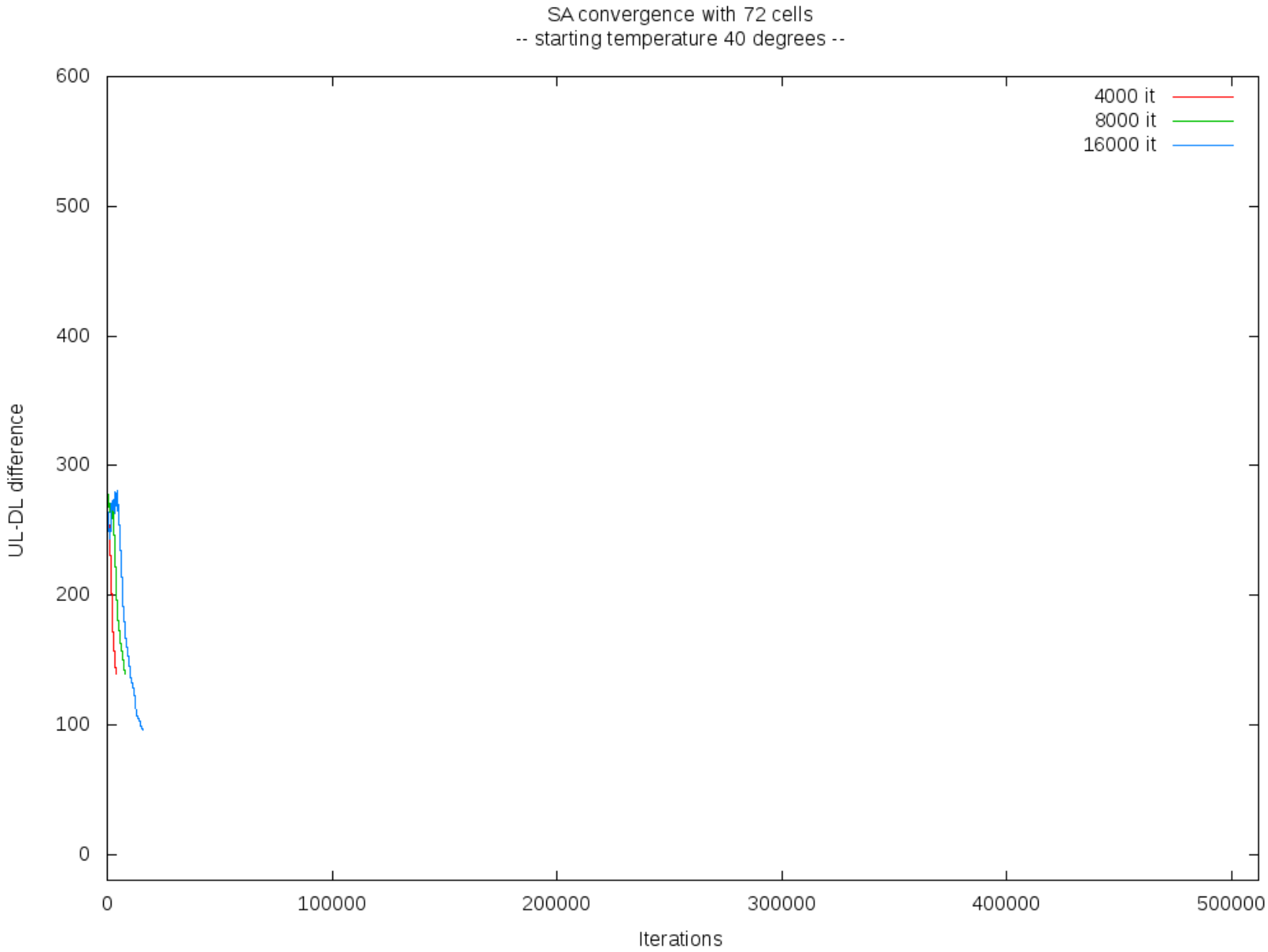
SA: Number of iterations



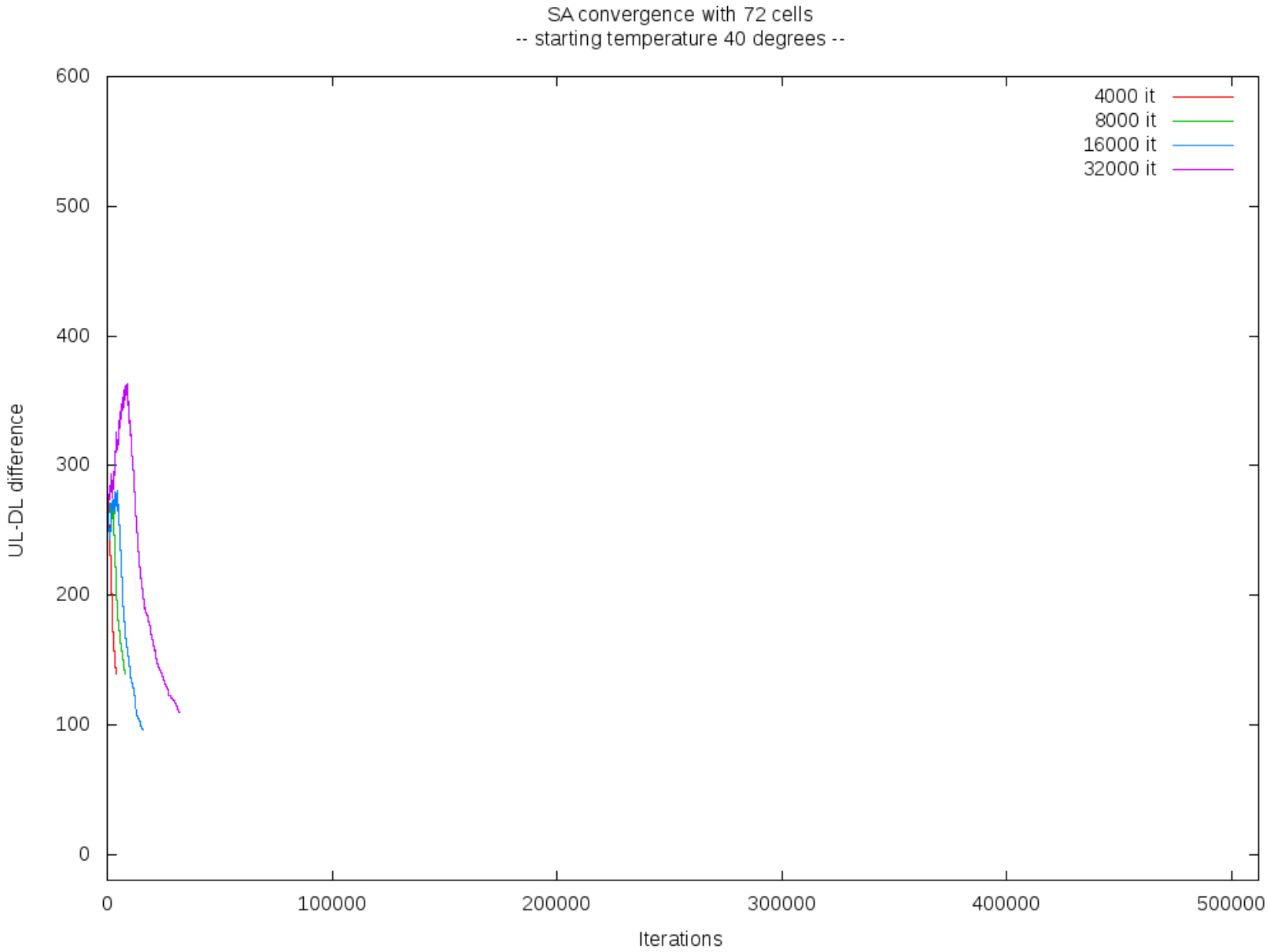
SA: Number of iterations



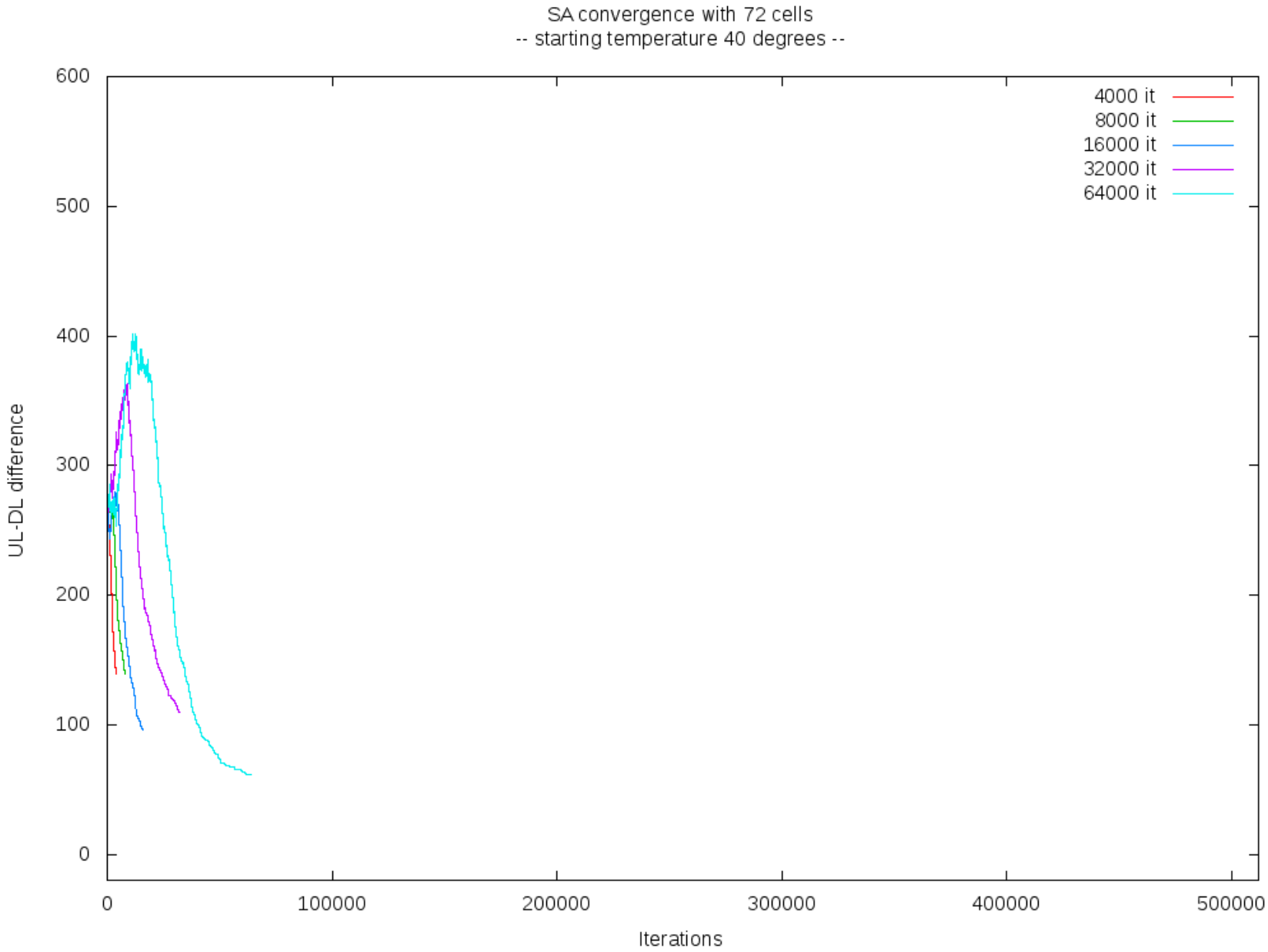
SA: Number of iterations



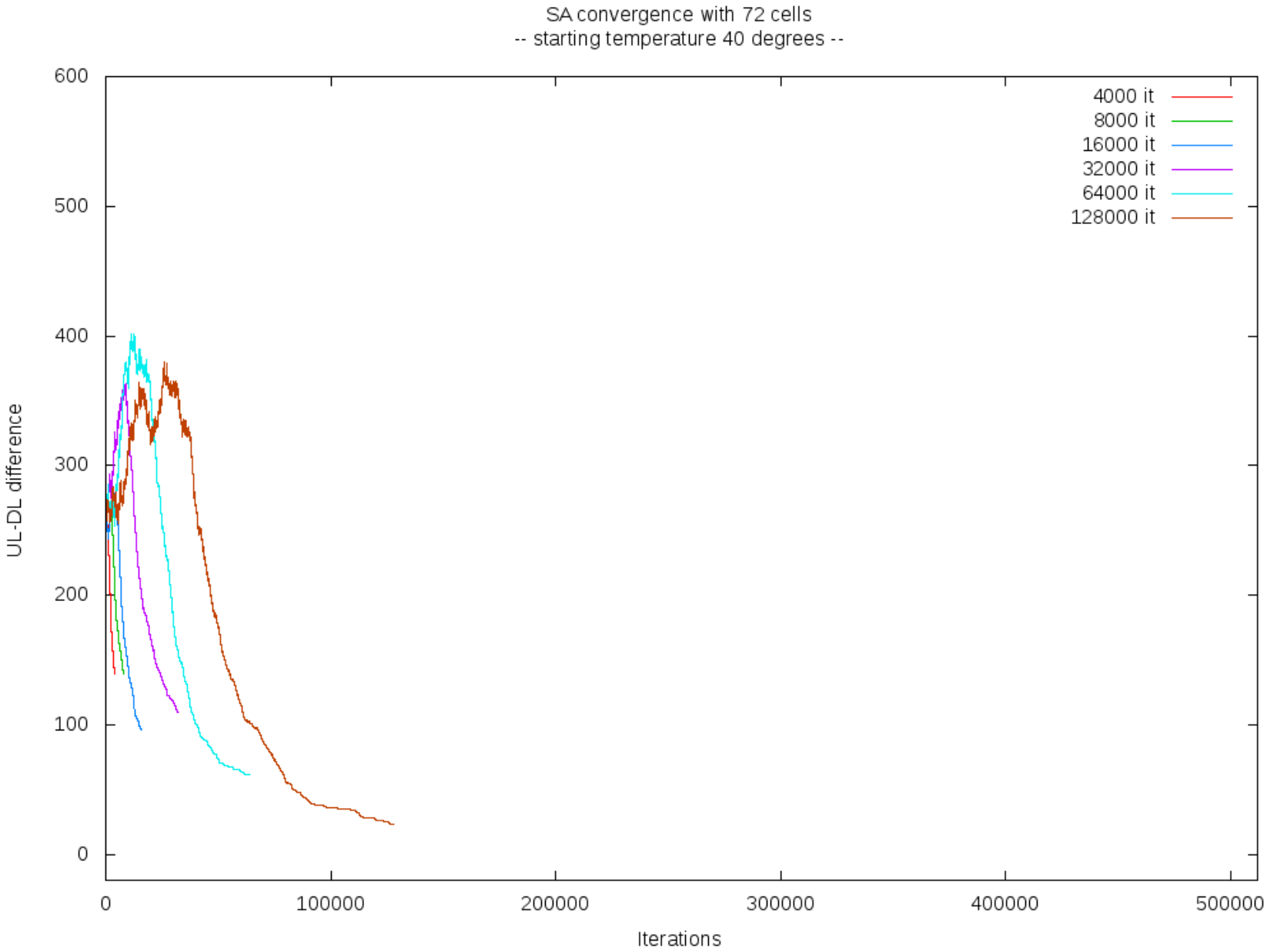
SA: Number of iterations



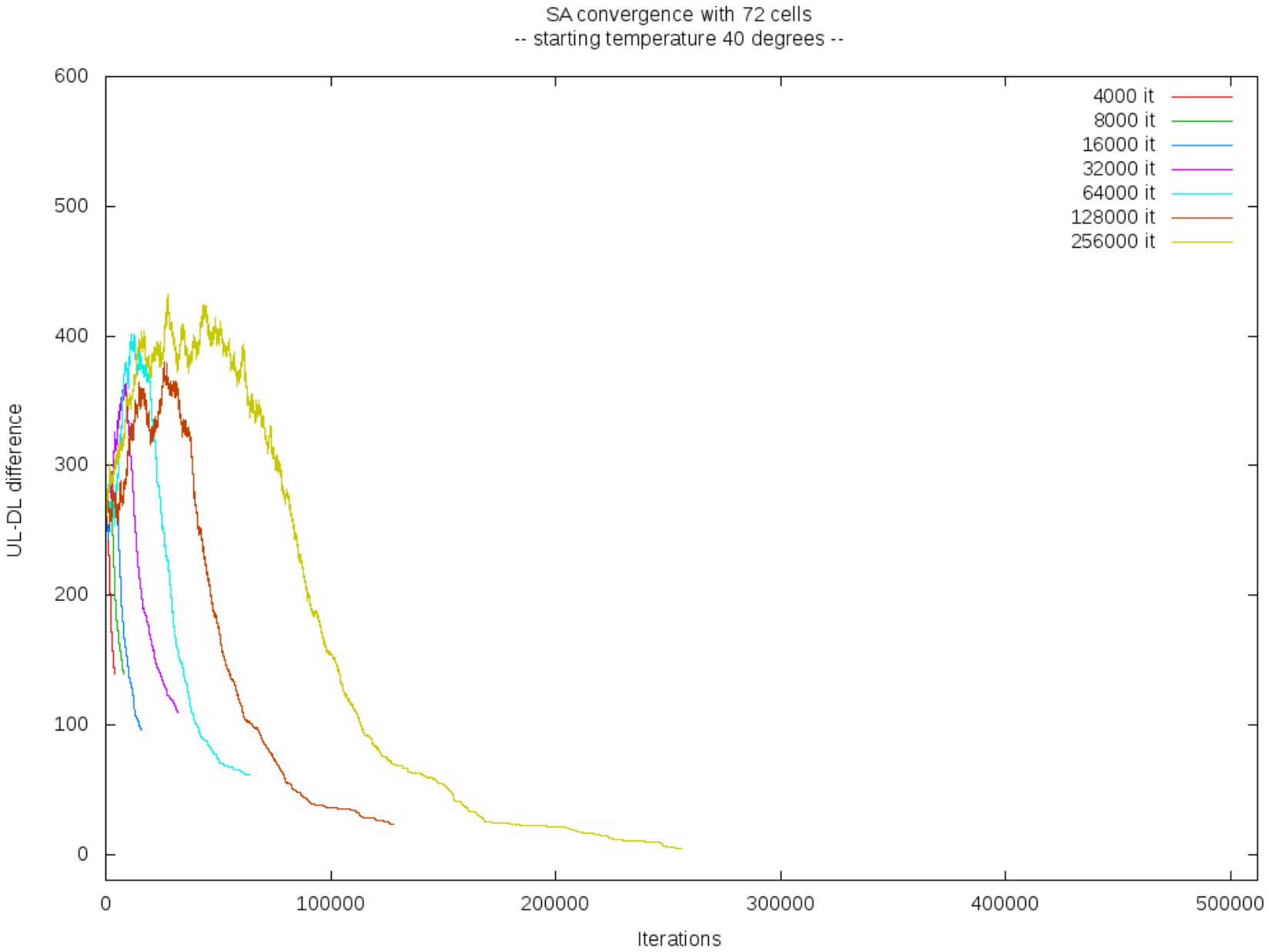
SA: Number of iterations



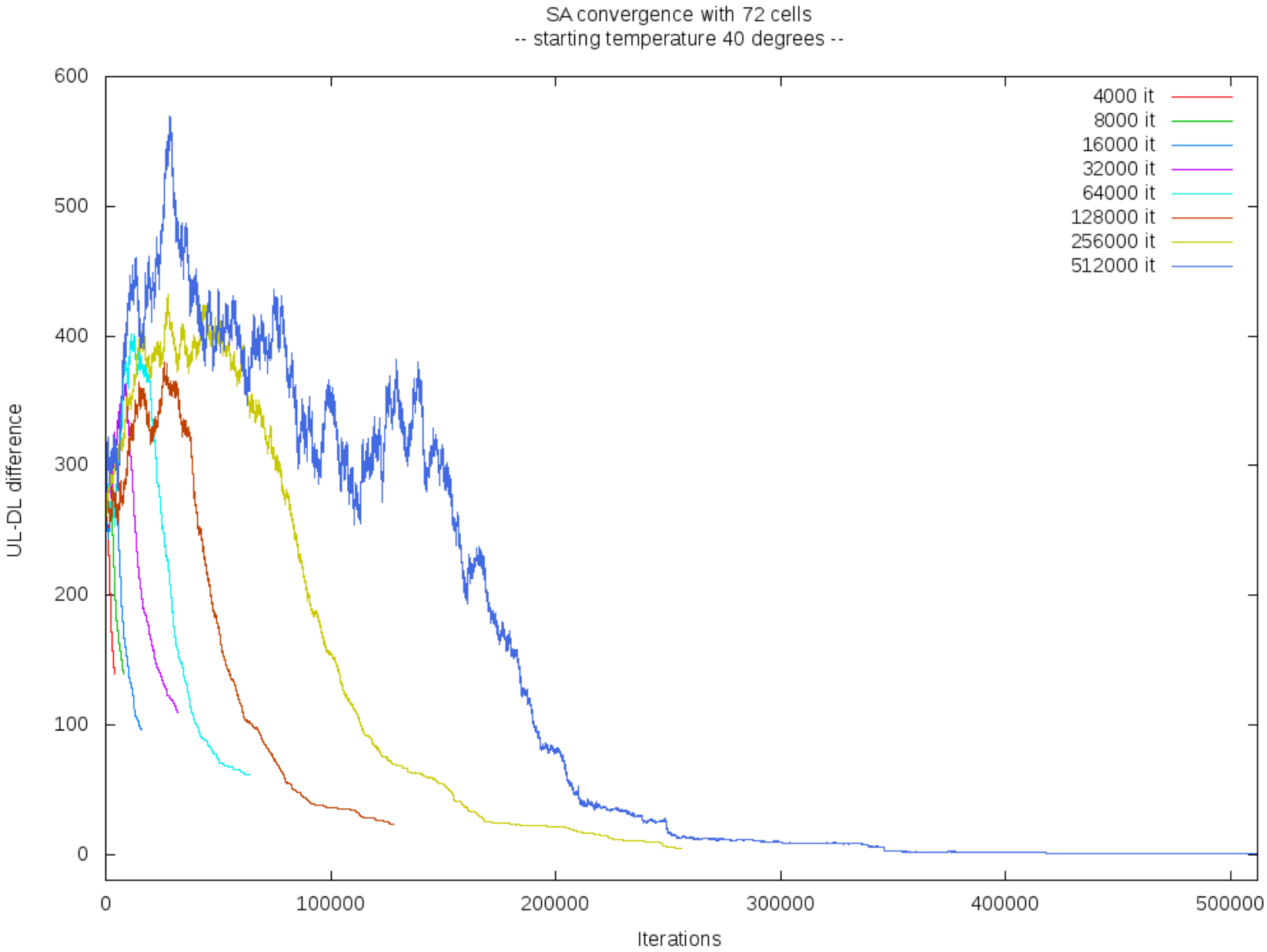
SA: Number of iterations



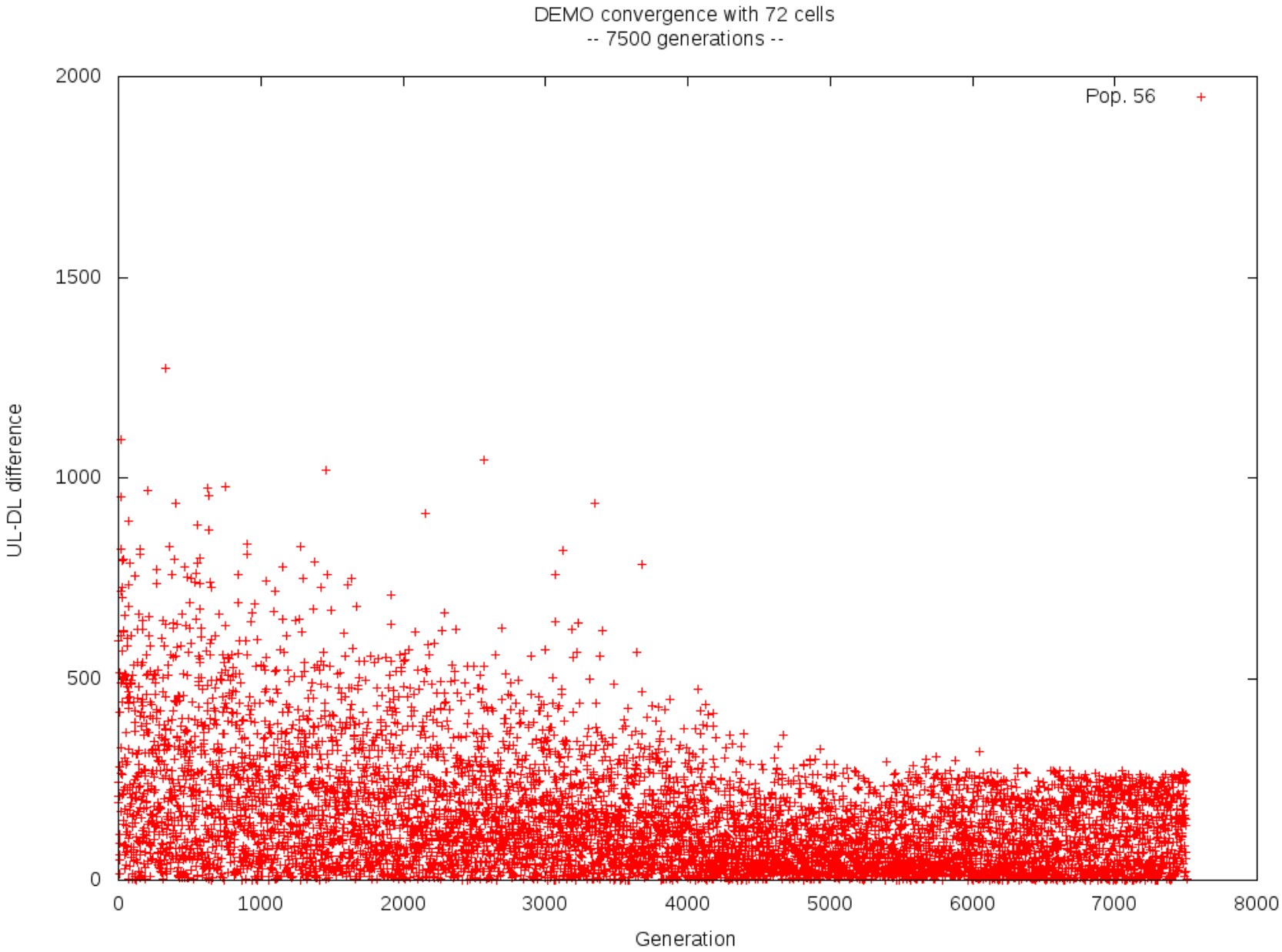
SA: Number of iterations



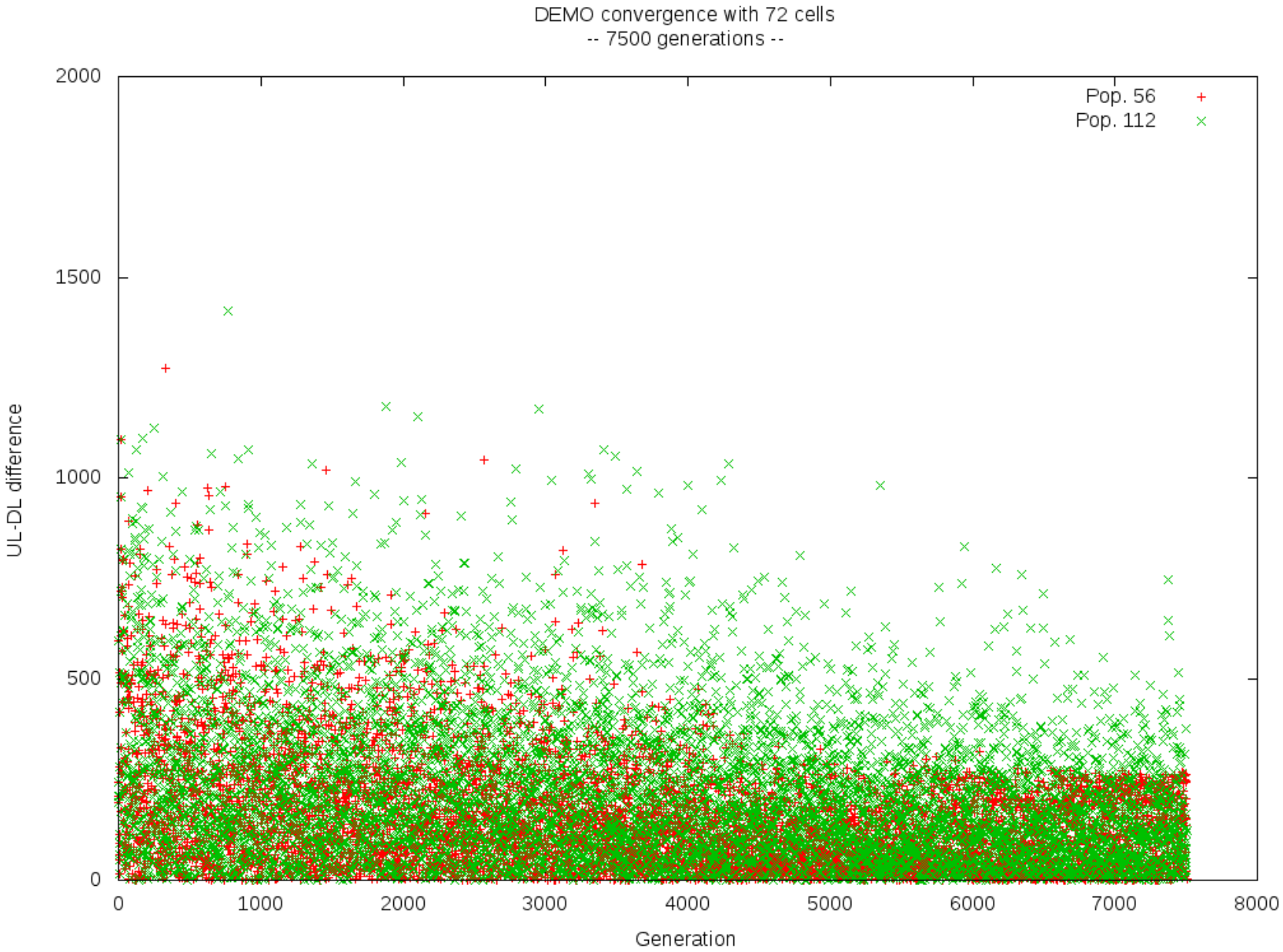
SA: Number of iterations



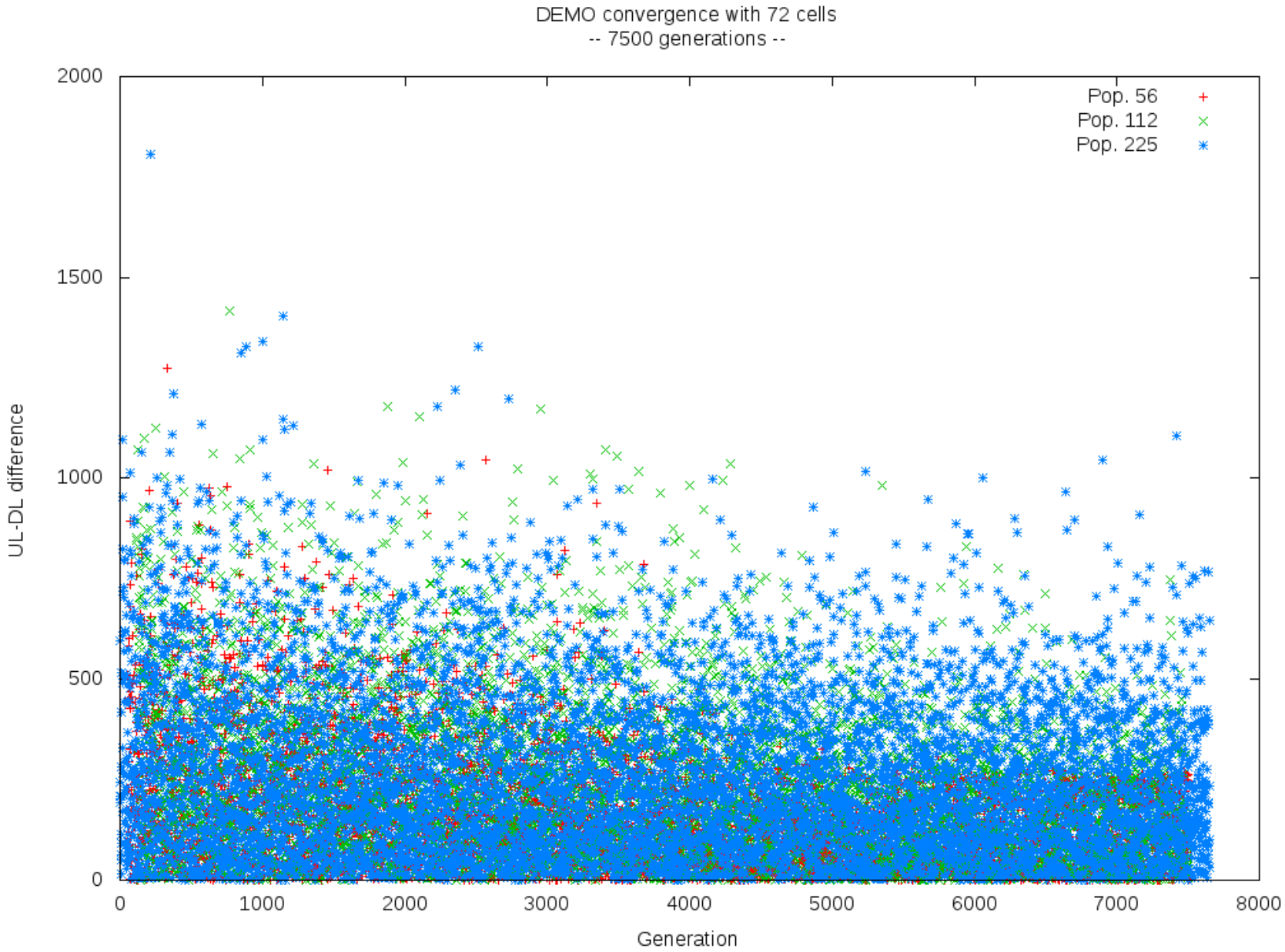
DEMO: Population size



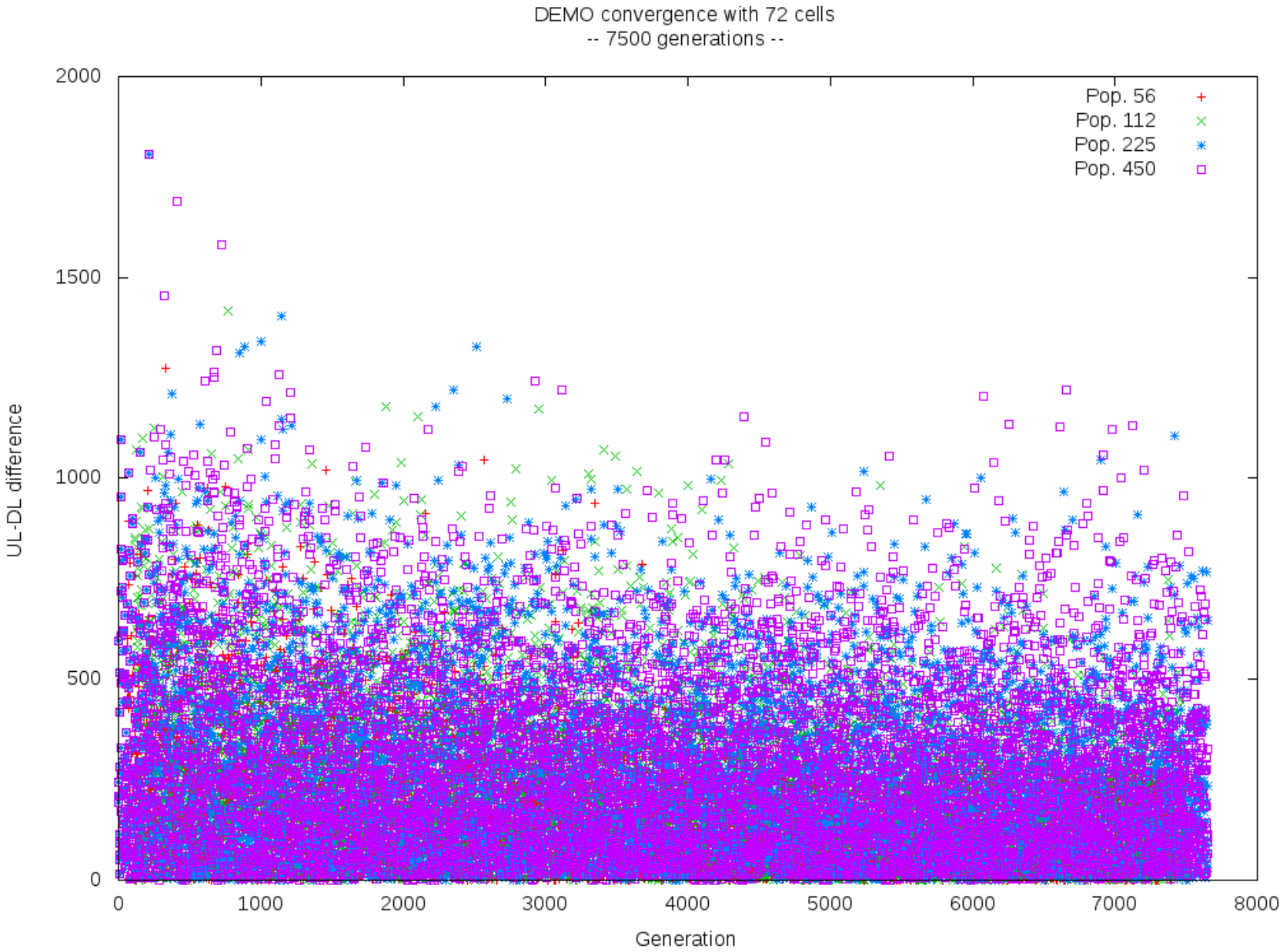
DEMO: Population size



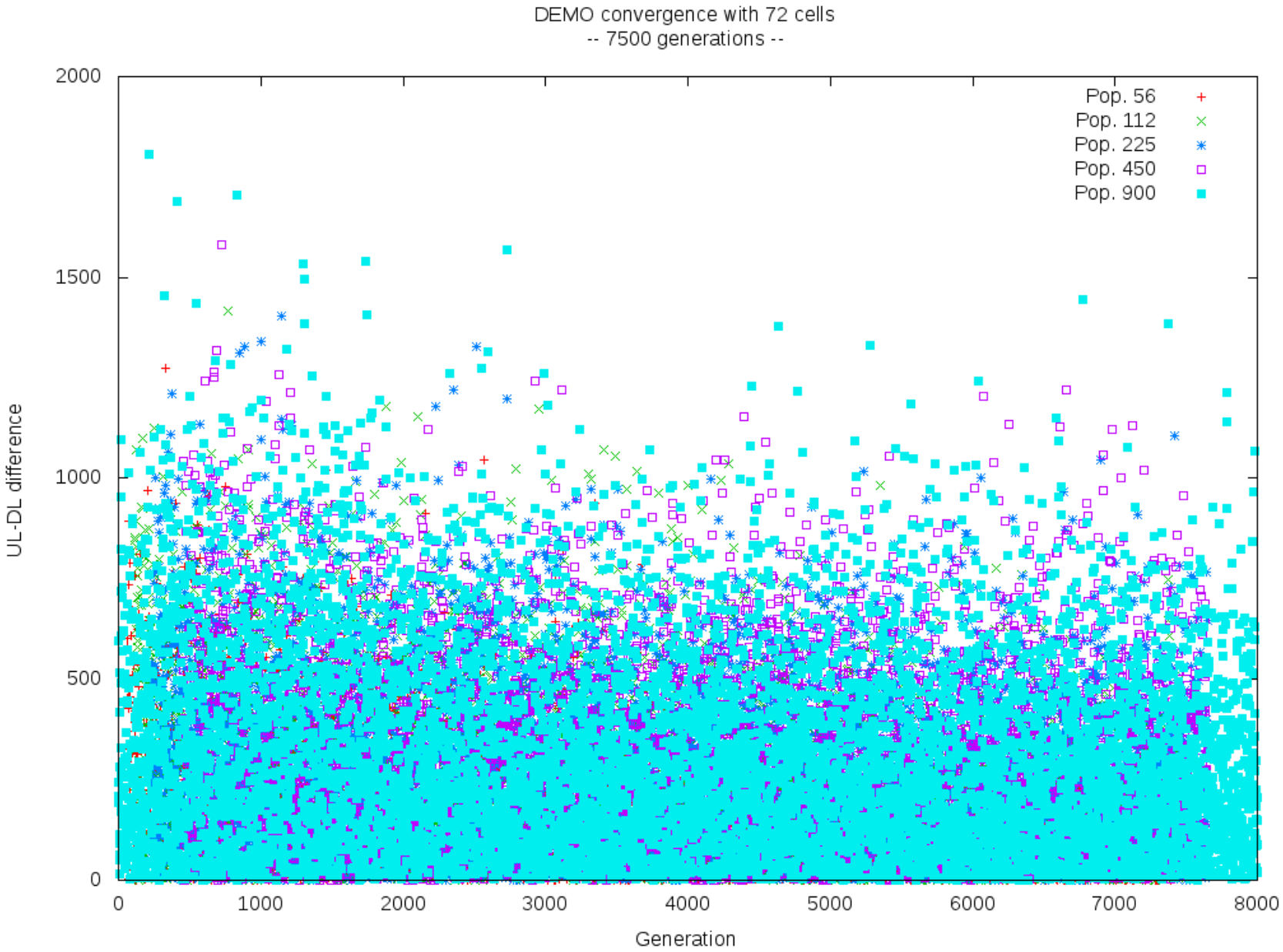
DEMO: Population size



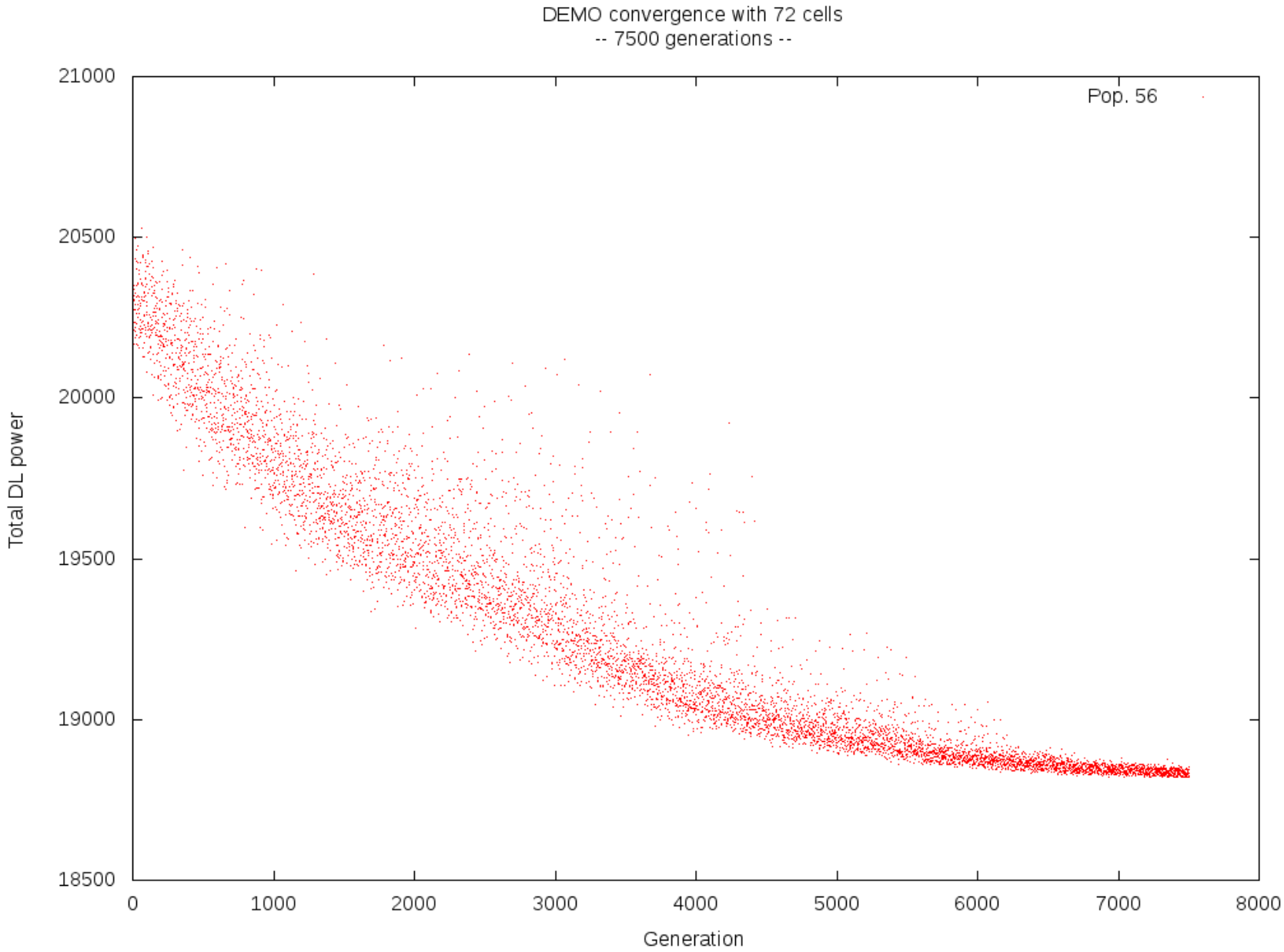
DEMO: Population size



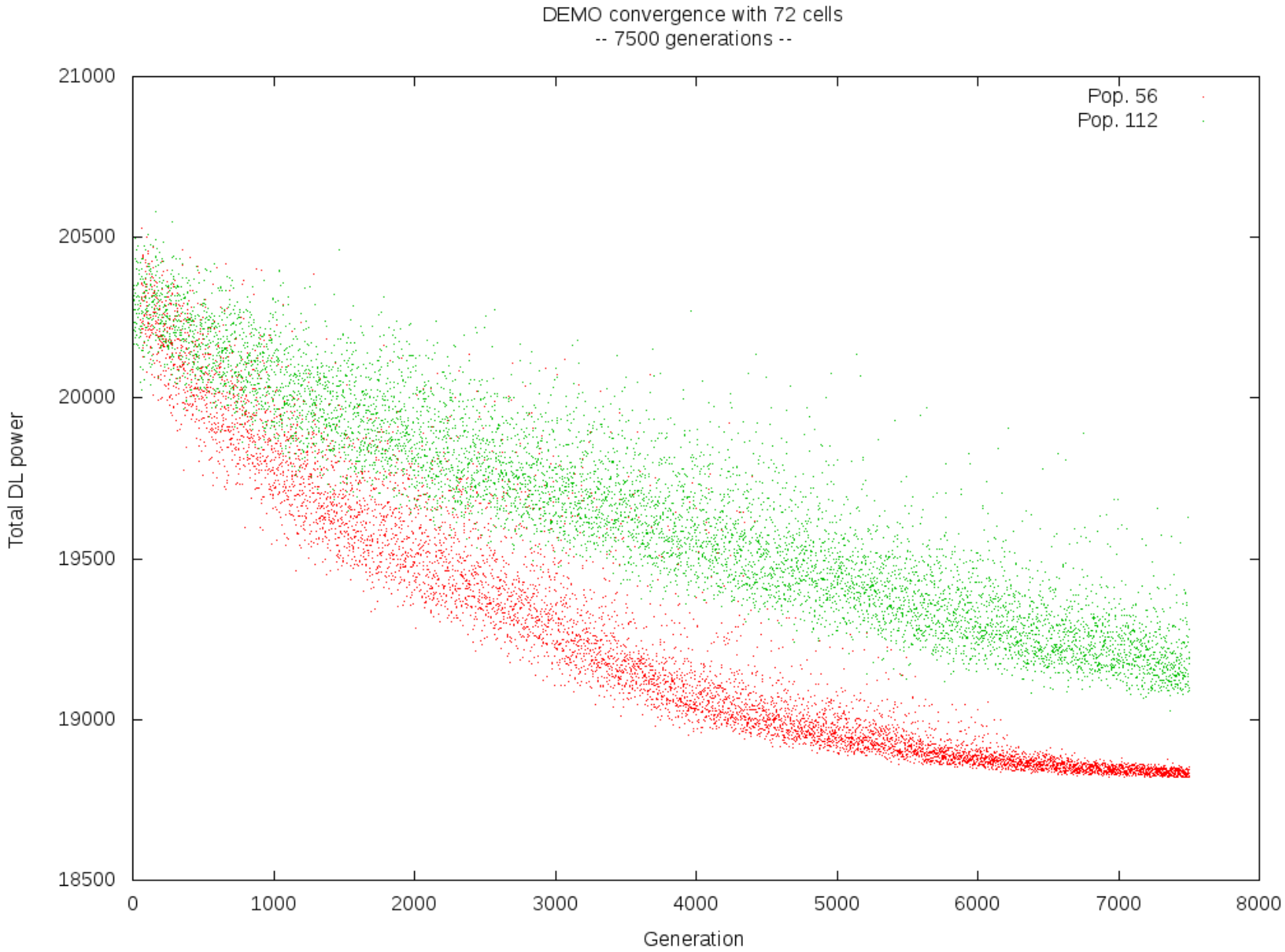
DEMO: Population size



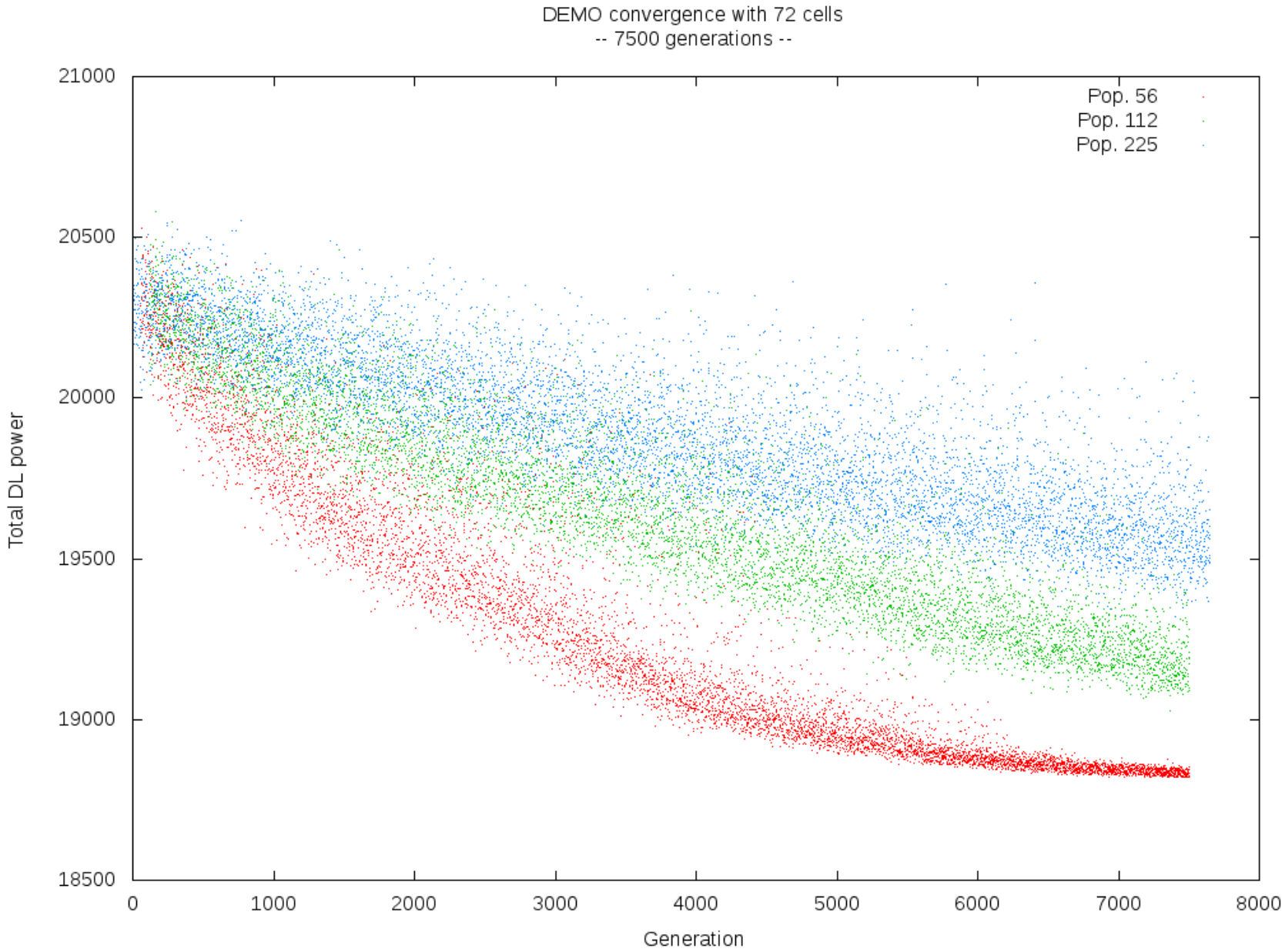
DEMO: Population size



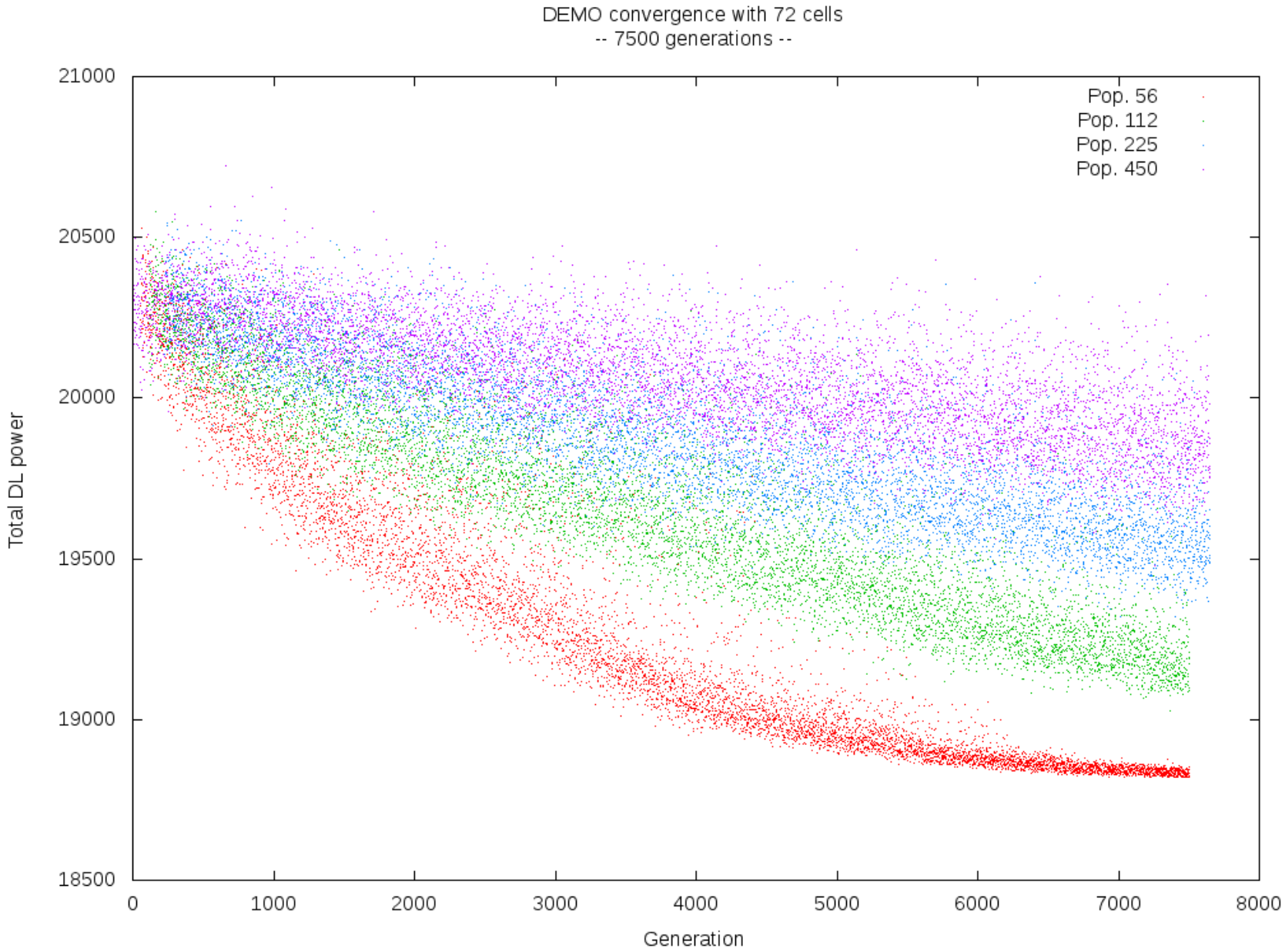
DEMO: Population size



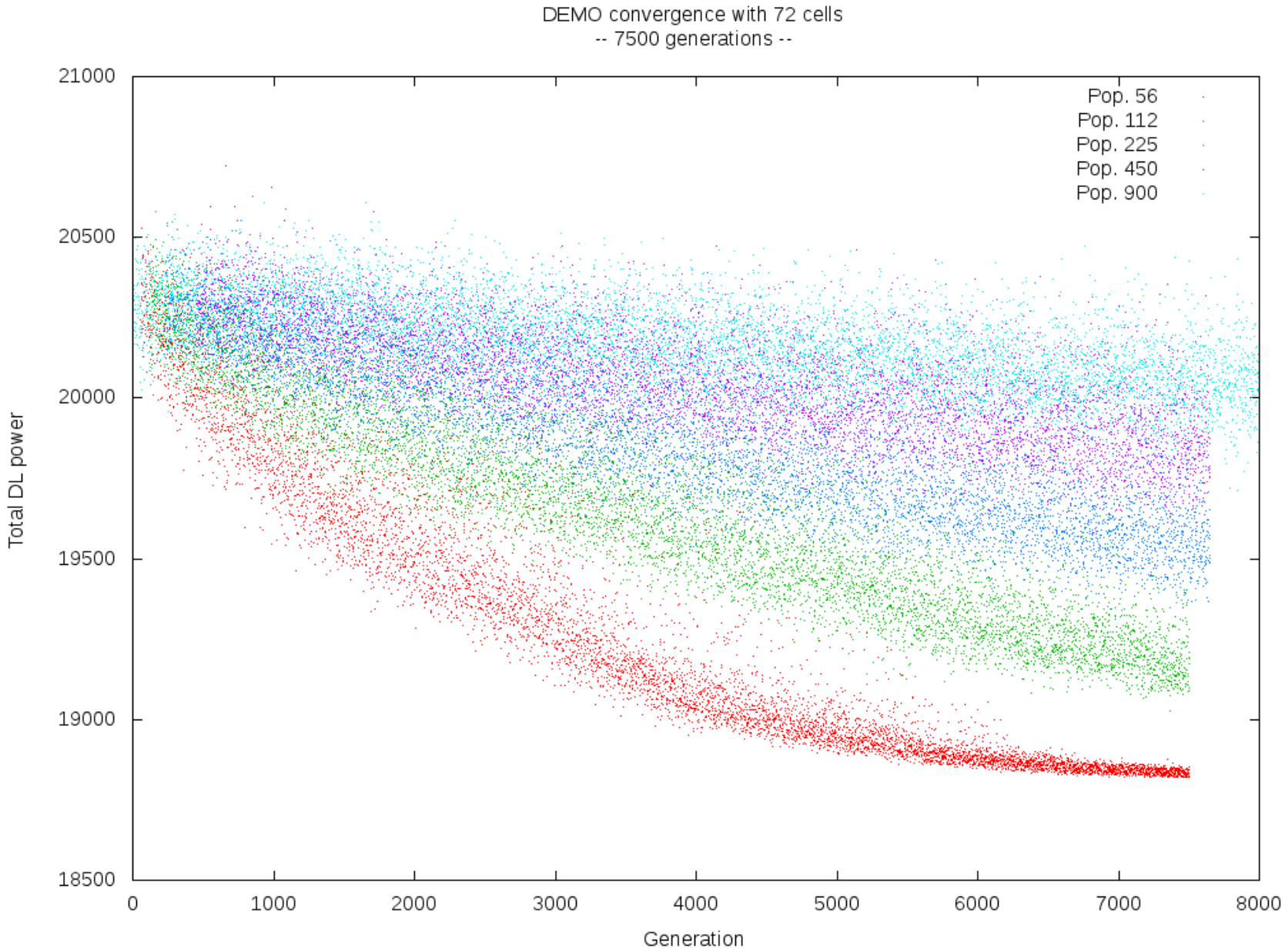
DEMO: Population size



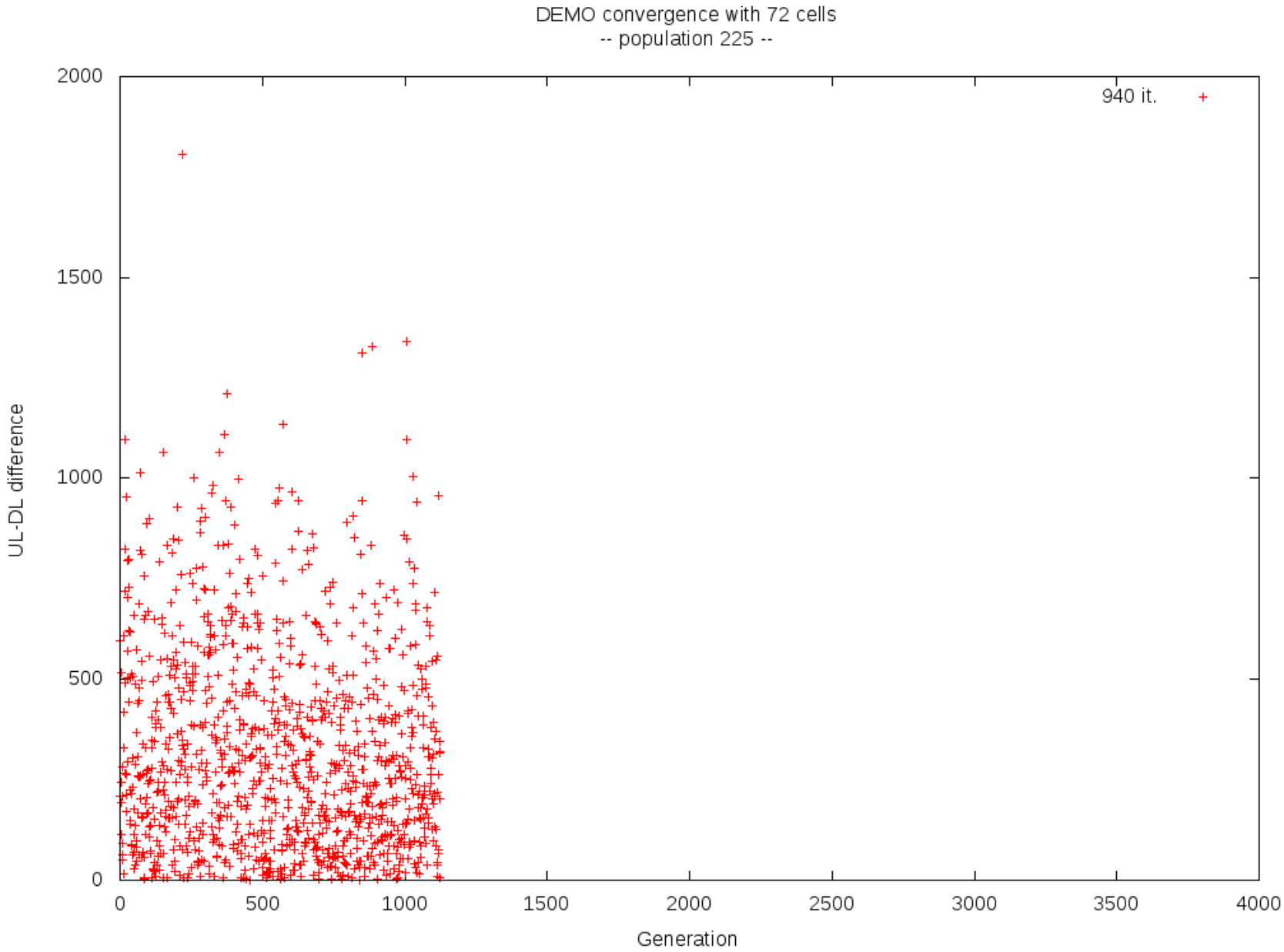
DEMO: Population size



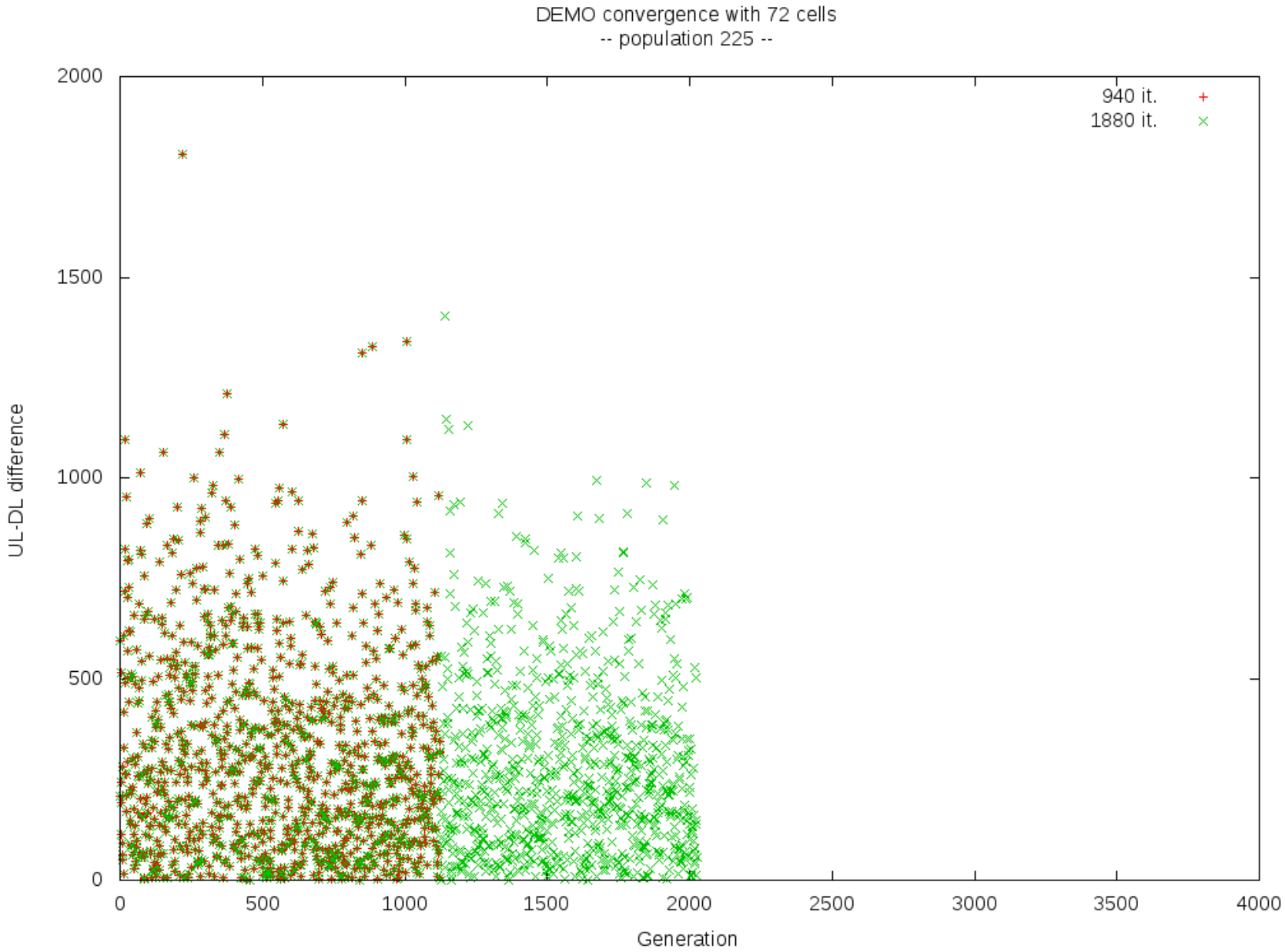
DEMO: Population size



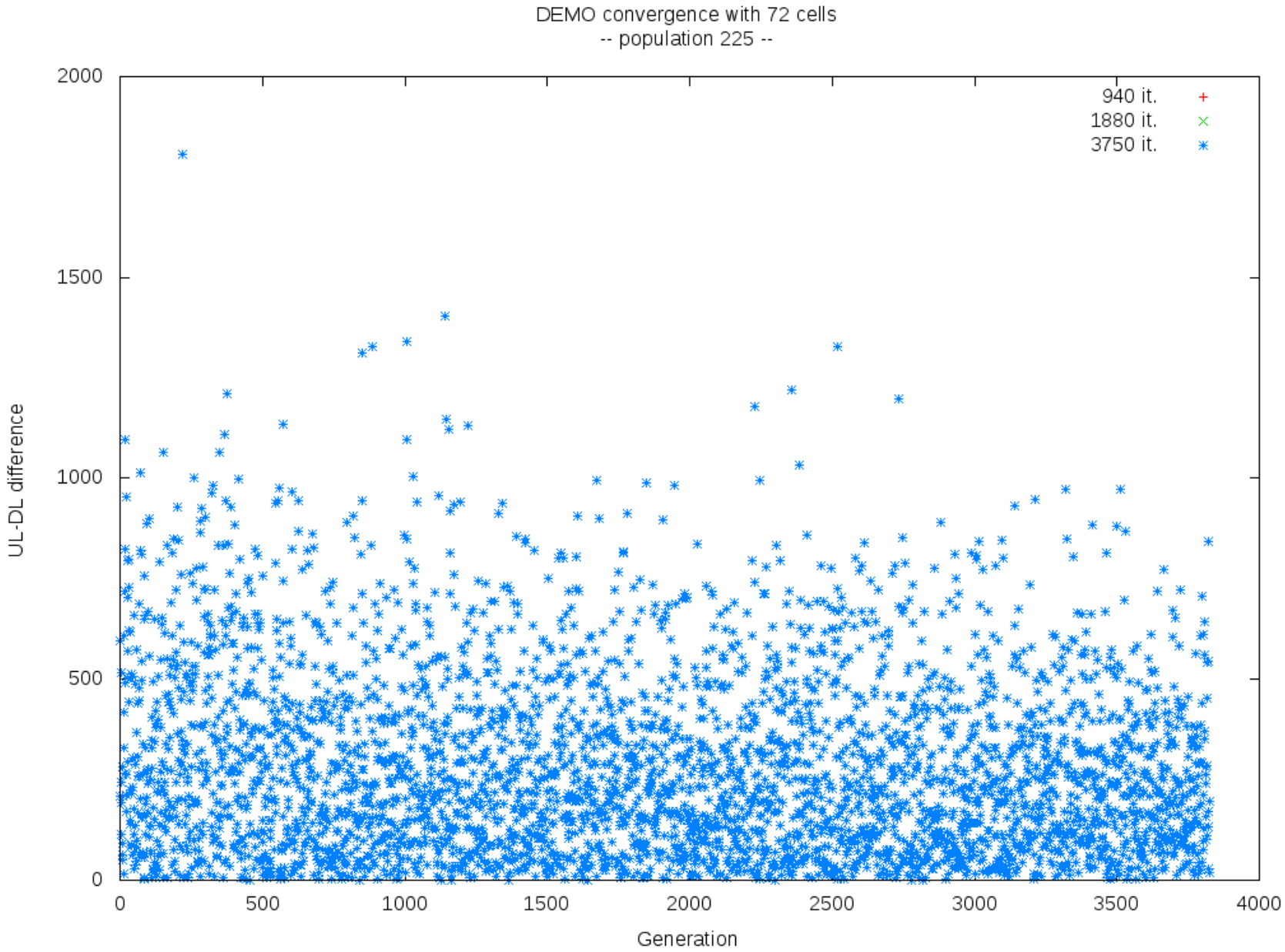
DEMO: Number of generations



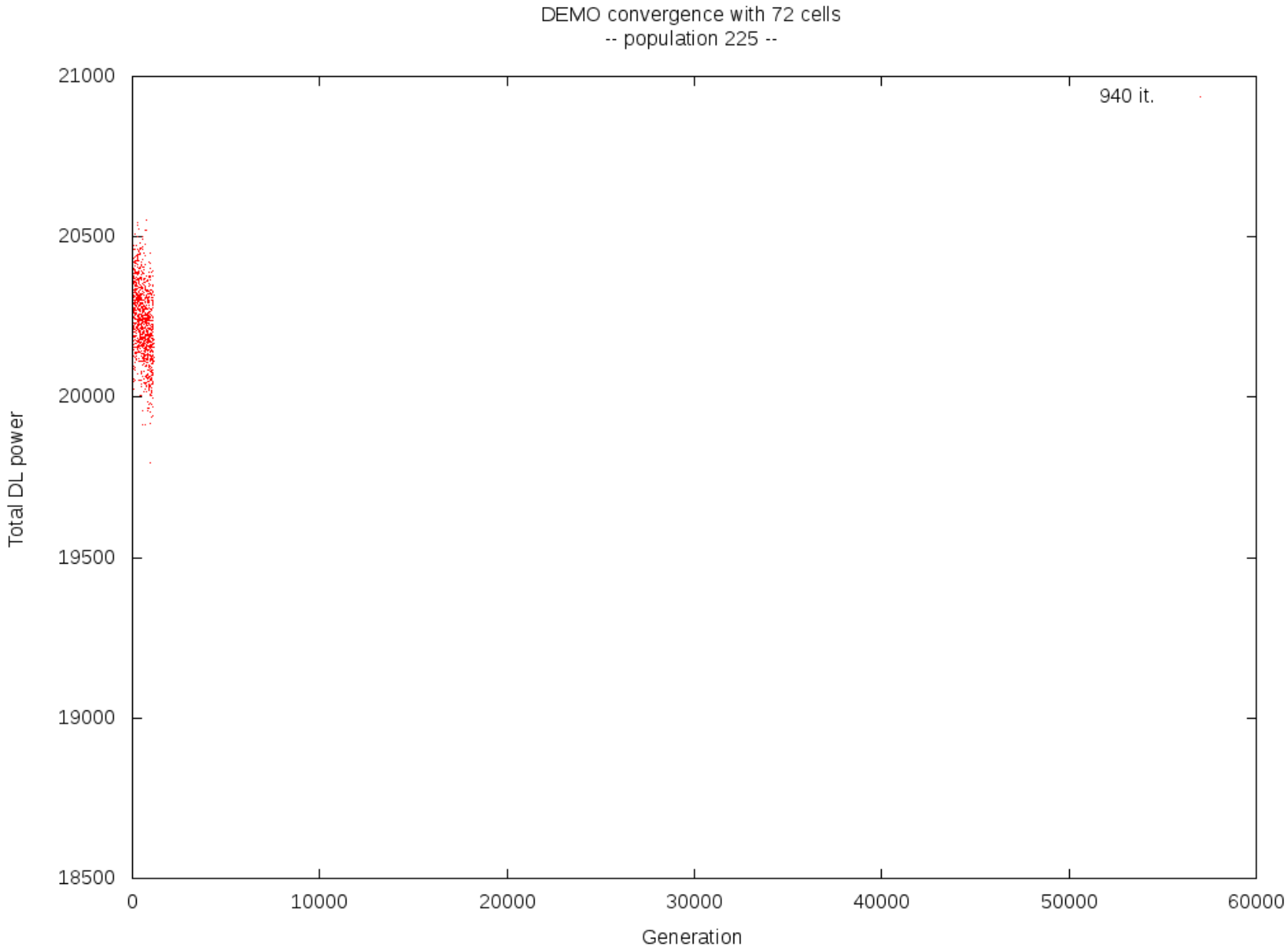
DEMO: Number of generations



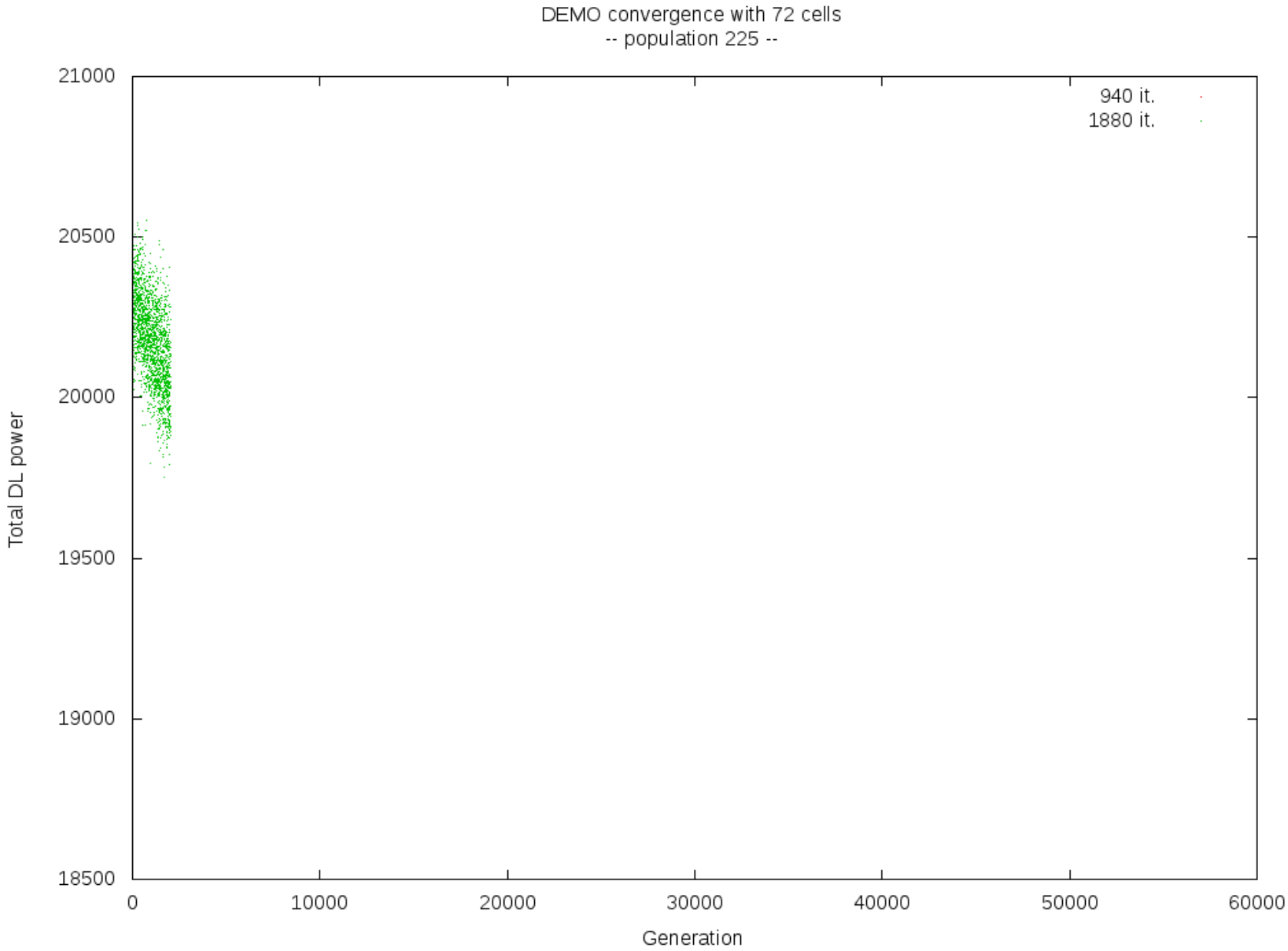
DEMO: Number of generations



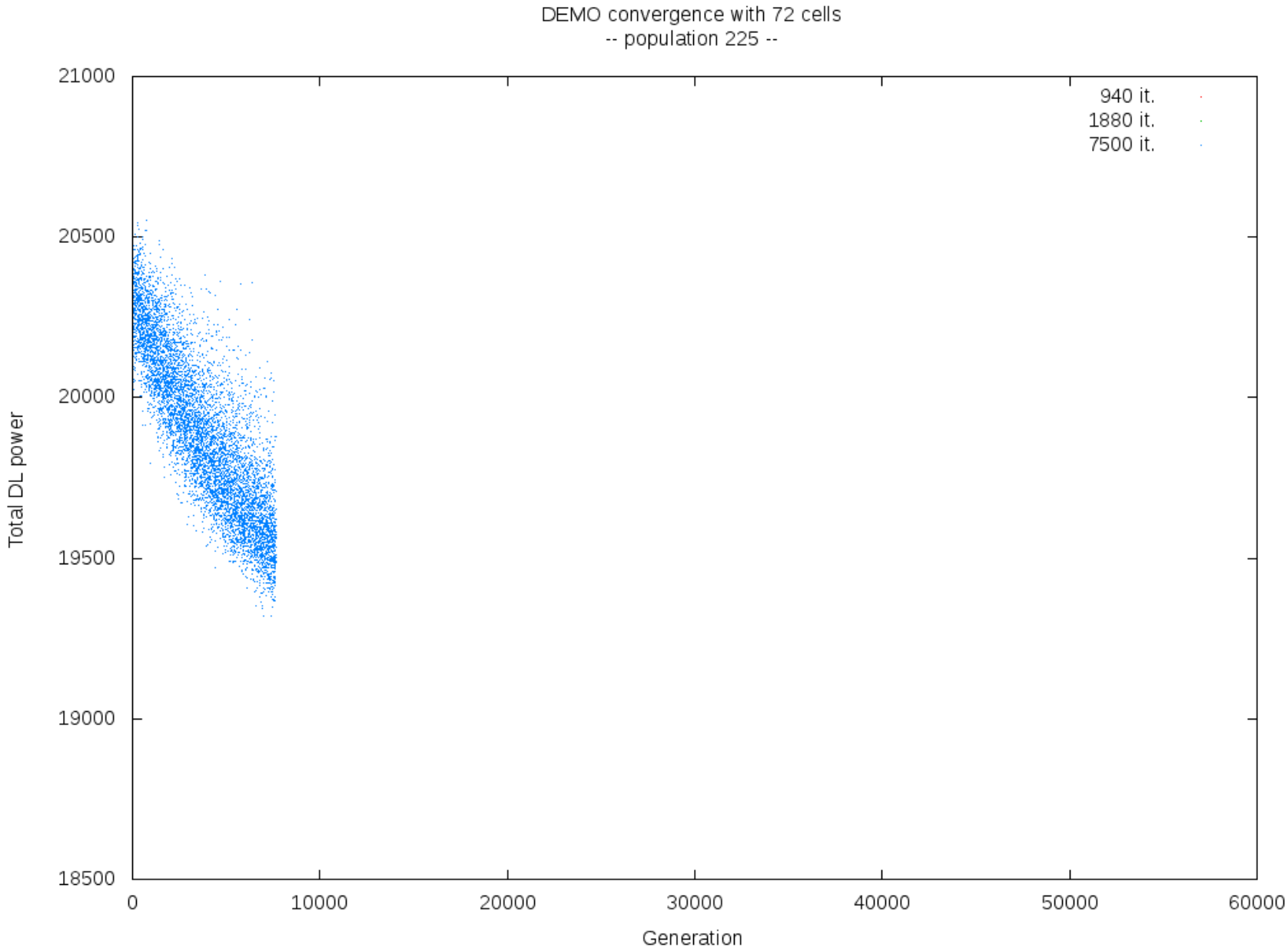
DEMO: Number of generations



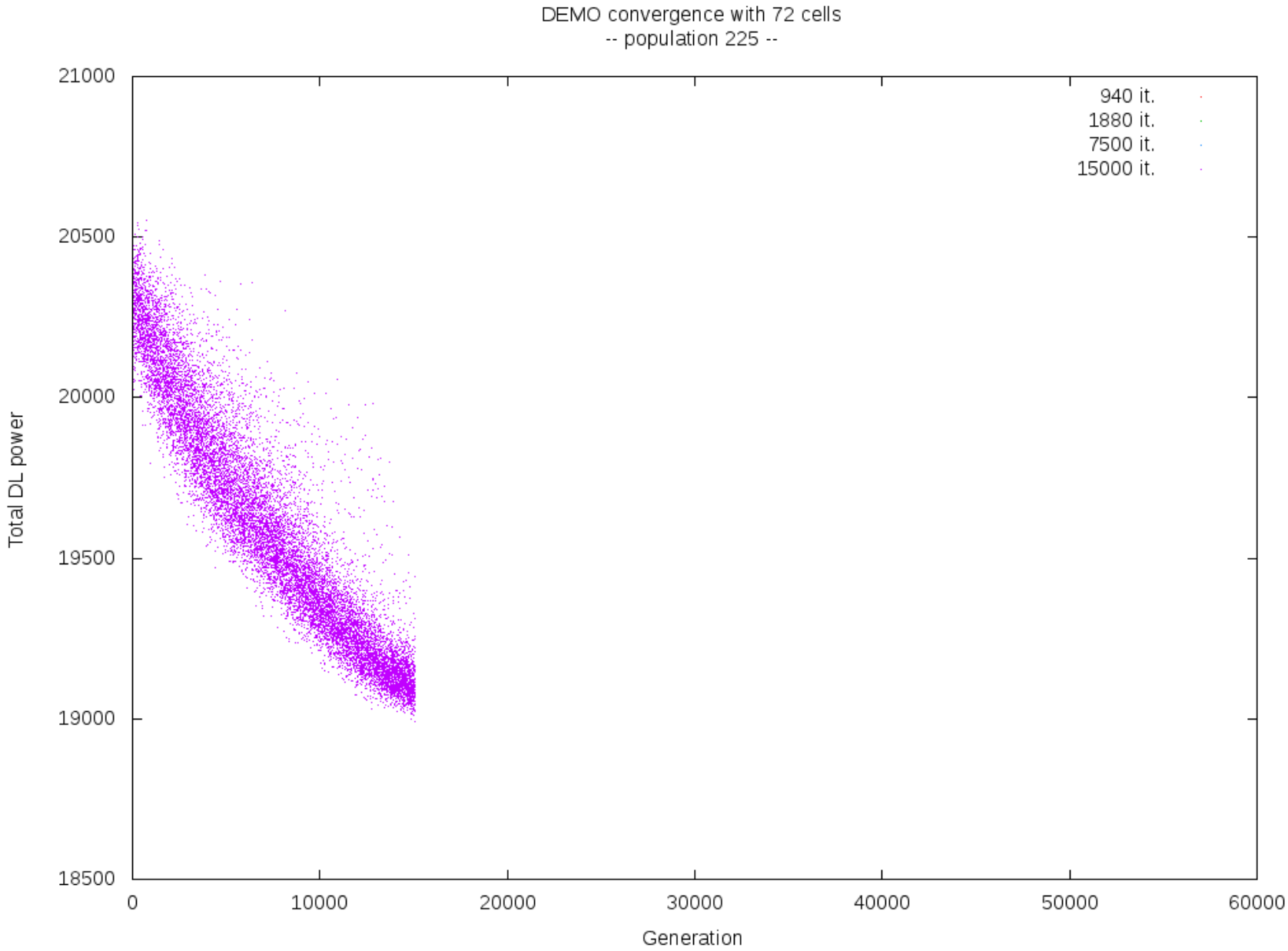
DEMO: Number of generations



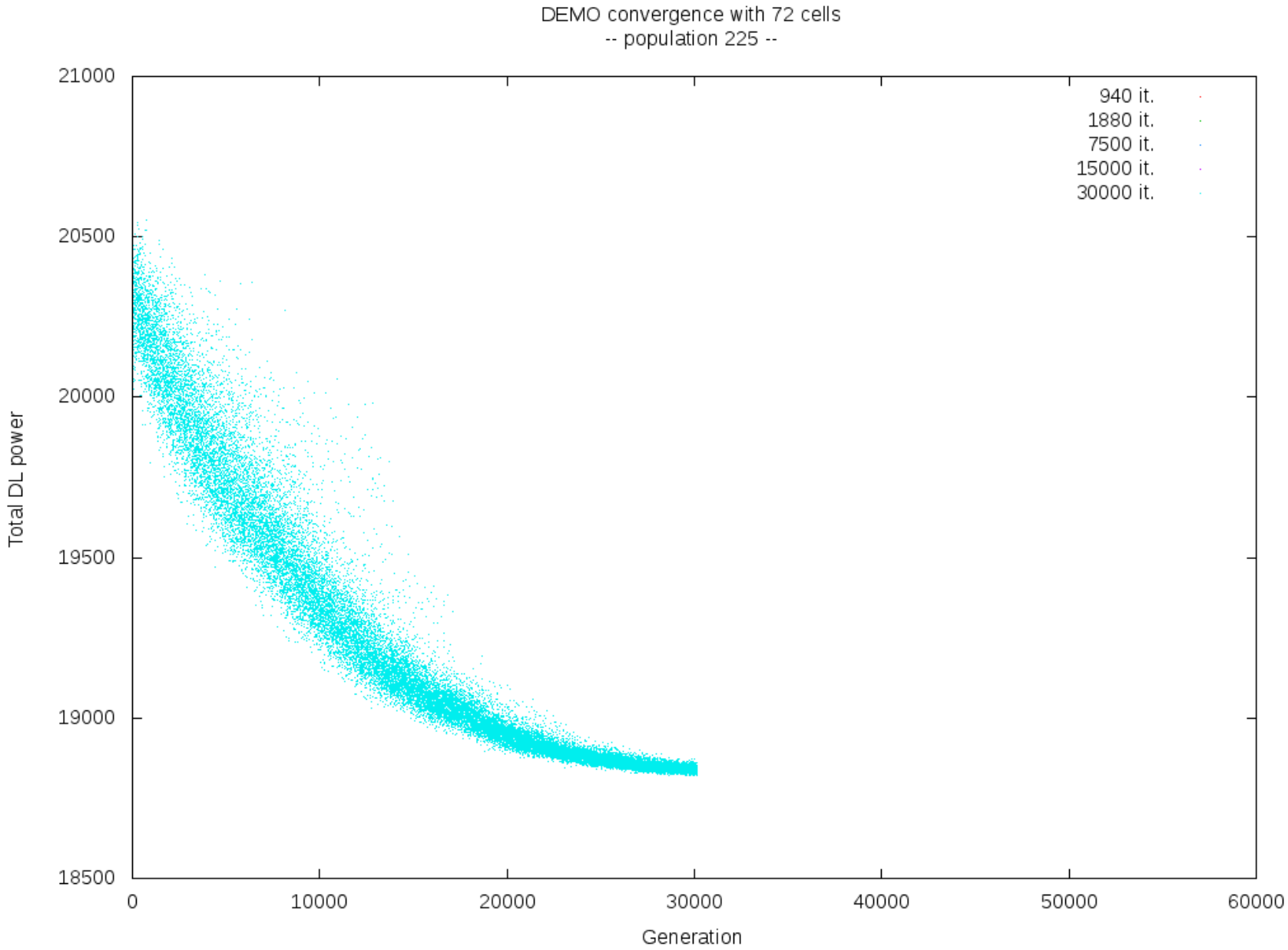
DEMO: Number of generations



DEMO: Number of generations



DEMO: Number of generations



DEMO: Number of generations