Genetic algorithm solution of the IHCP using parallel computing and commercial CHT software

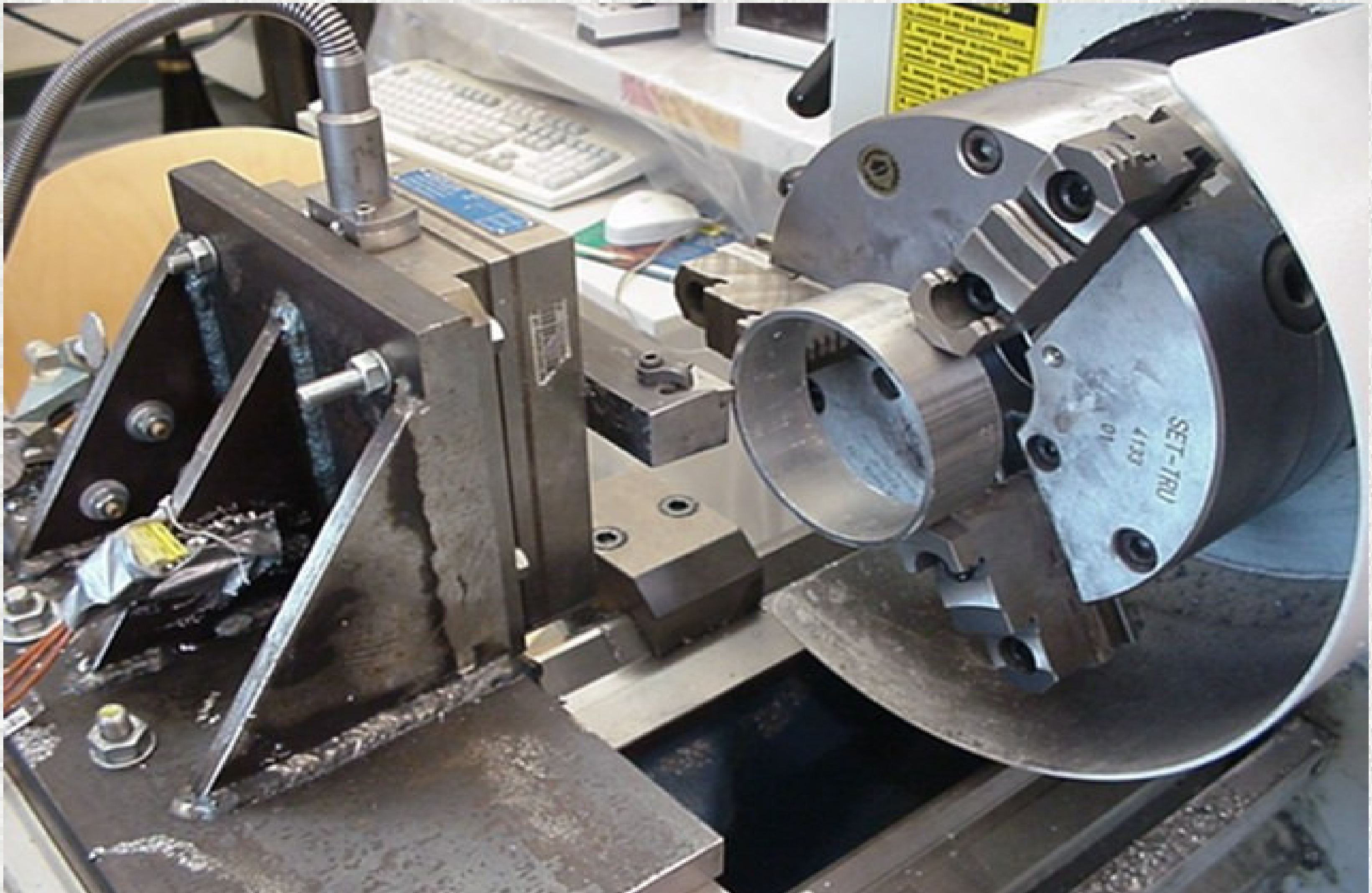
Keith A. Woodbury
Suprasanna Duvvuri
Mechanical Engineering Department

THE UNIVERSITY OF
ALABAMA
ENGINEERING

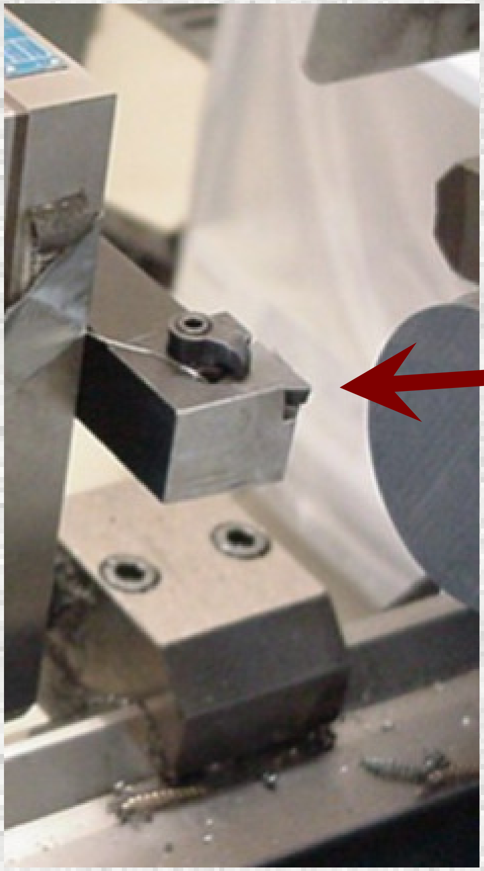
Outline

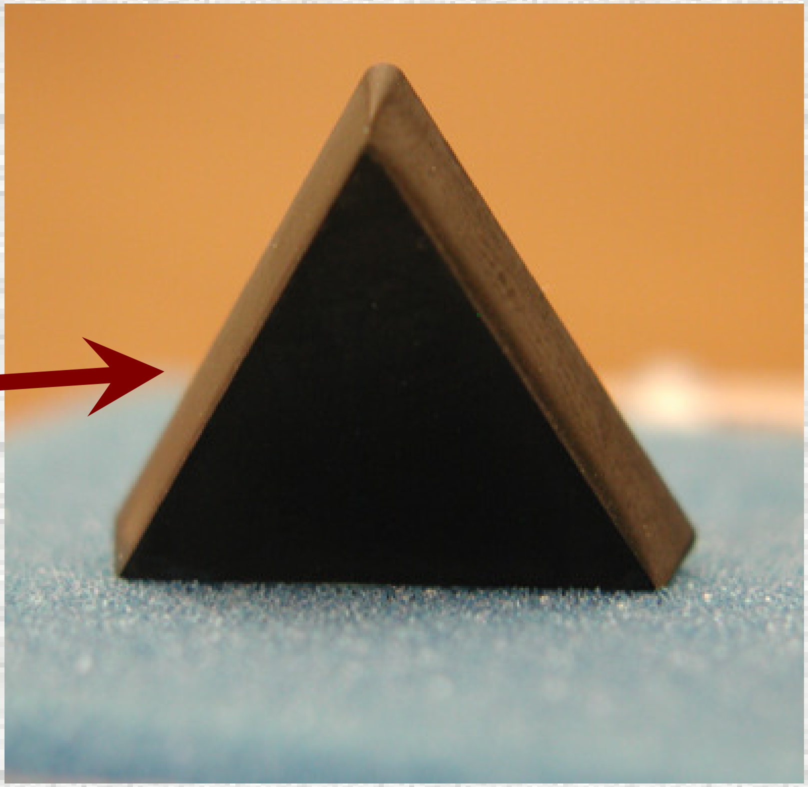
- Motivation – Machining application
- Genetic Algorithms
- Parallel Computing
- Test problem – 1D IHCP
- Results
- Conclusions

Motivation

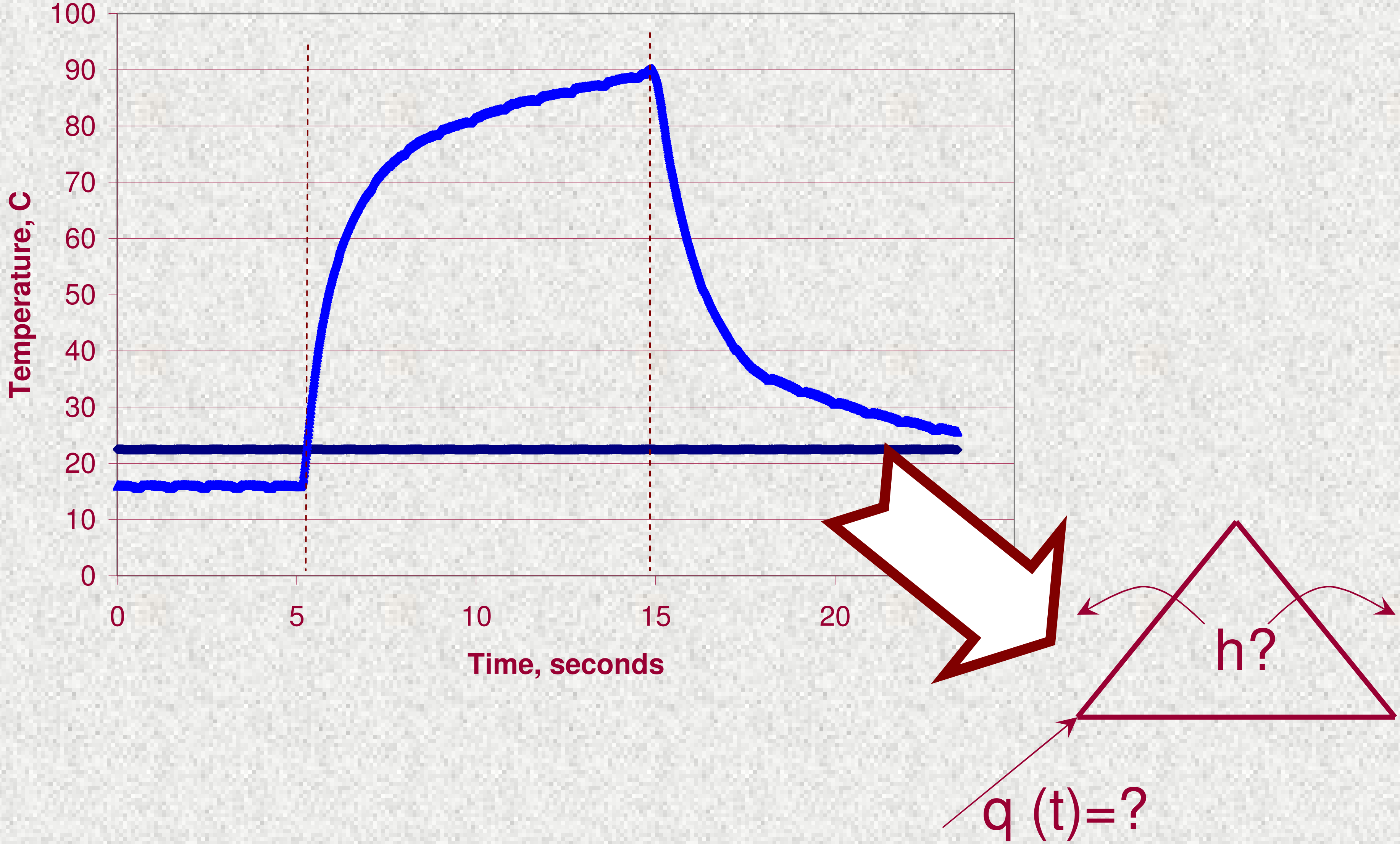


Tool Geometry





Inverse Problem



Approach

- Use genetic algorithms to solve the optimization problem
- Use commercial software (FLUENT) to solve forward heat conduction problem
- Parallelize The genetic algorithm code to reduce the computational time

Genetic Algorithms

Based on mechanics of natural selection and natural genetics

1) Selection - Individual strings are chosen according to their fitness values
2) Reproduction – selected strings mated at random to produce strings with better fitness
3) Mutation – this introduces information into the solution that was not present in the initial population

Genetic Algorithms

- Coded parameters (contrast with evolutionary algorithms)
- Fundamental Theorem of Genetic Algorithms: short, low-order, above average schemata receive exponentially increasing trials in subsequent generations
- Implicit parallelism: when n structures are processed in each generation, a genetic algorithm processes n^3 schemata (building blocks)

Genetic Algorithm

- Maximizes, not minimizes
- Fitness function

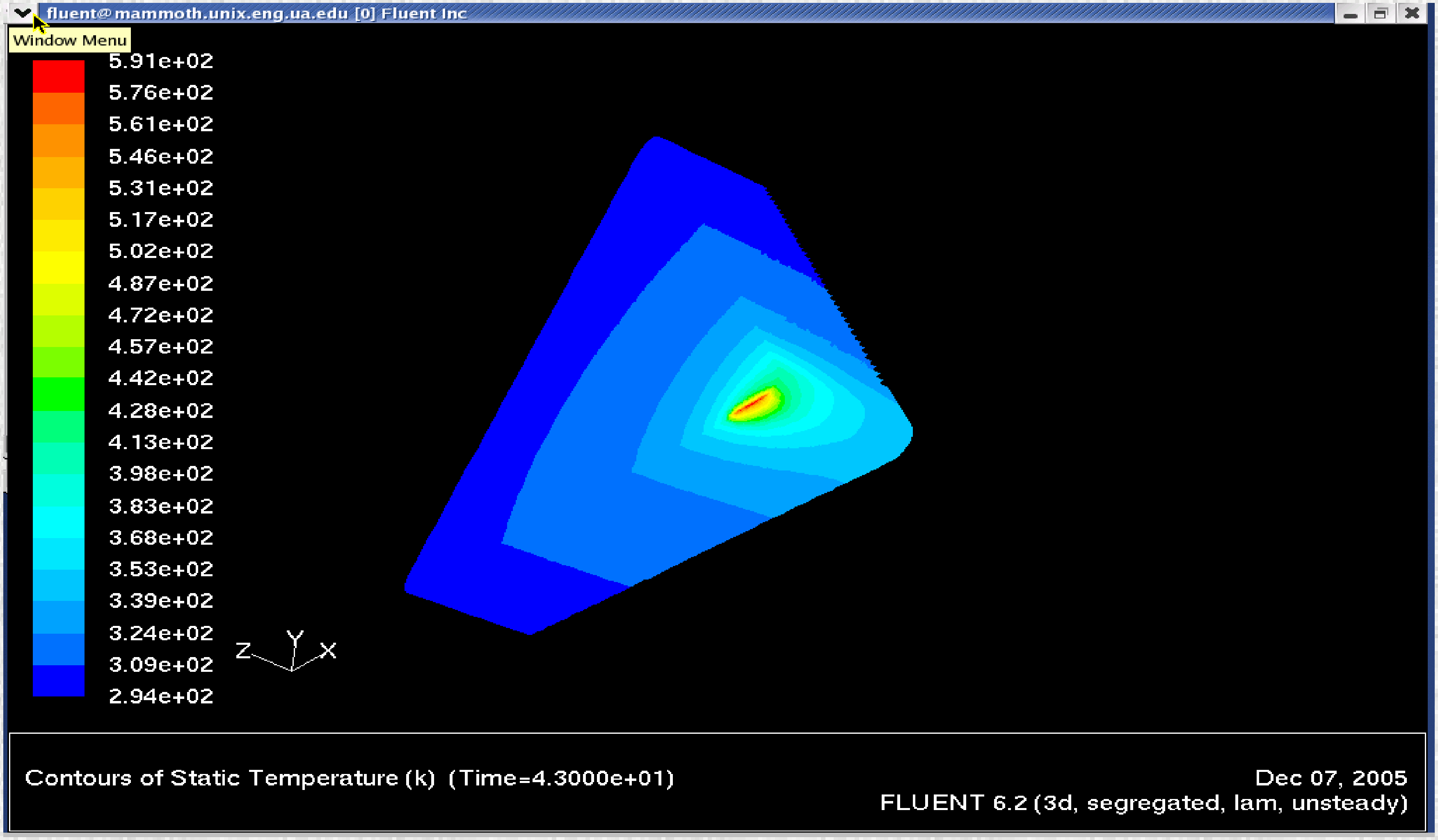
$$f = \frac{1}{1 + 10 error_{RMS}}$$

$$error_{RMS} = \sqrt{\frac{\sum_{i=1}^{n_{obs}} (Y_i - T_i)^2}{n_{obs}}}$$

Genetic Algorithm

- We adapted Goldberg's "SGA" algorithm
  - Standard implementation
  - Many configurable options

FLUENT simulations



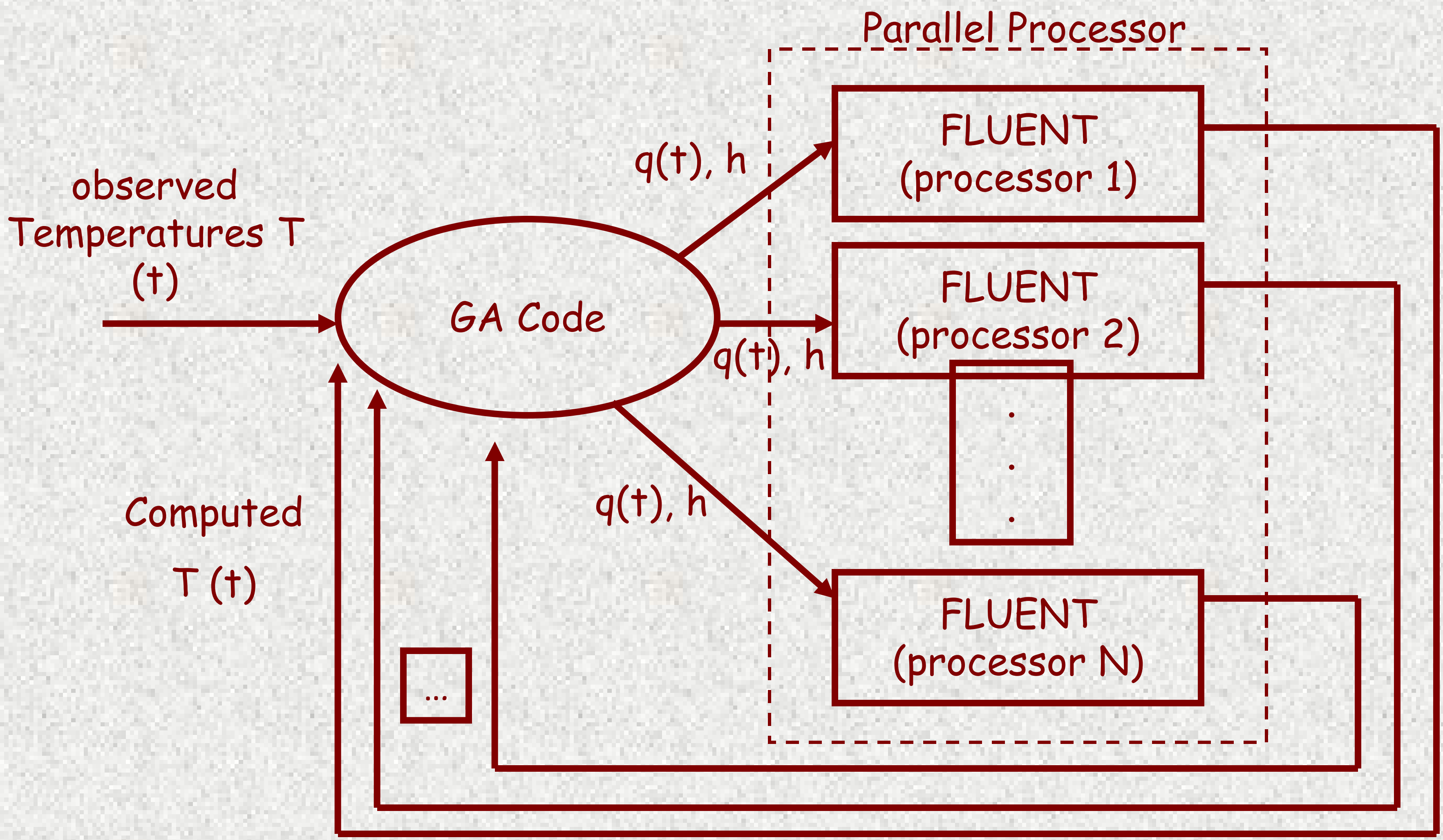
Parallel Computing

- "Embarrassingly" Parallel
  - Each processor runs a single, independent FLUENT simulation
  - Speedup time scales directly with number of processors

Parallel Computing

- Platform
  - Mechanical Engineering Department
  - 8 node (16 CPU) Dell PowerEdge HPC
  - EMT64 (Xeon) processors @ 3.20 GHz
  - Theoretical throughput 102.4 Gigaflops

Parallel Computing



Test Problem

- Want to learn the best combination of many parameters to obtain a good solution on the tool 3D IHCP
- Study a simple well-known 1D IHCP

Test Problem

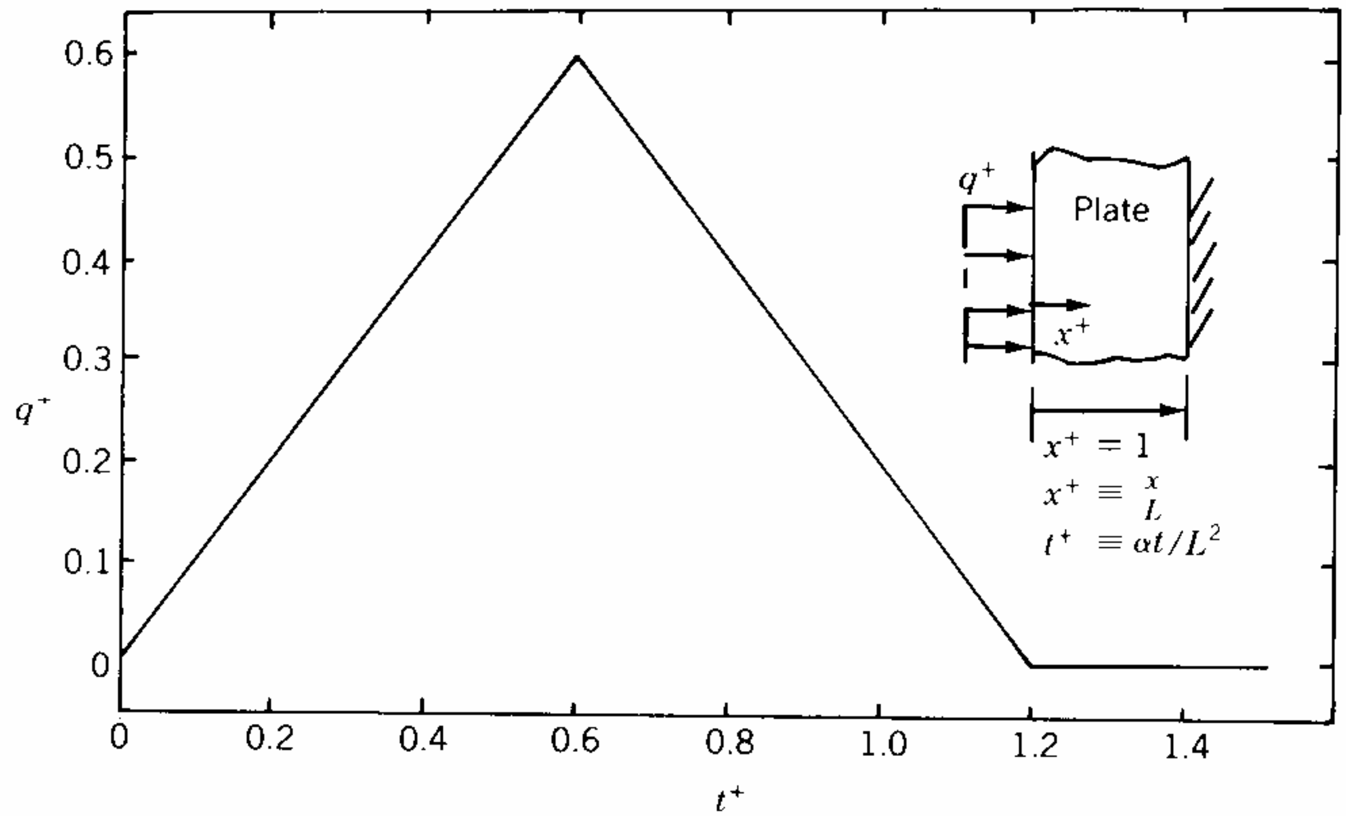
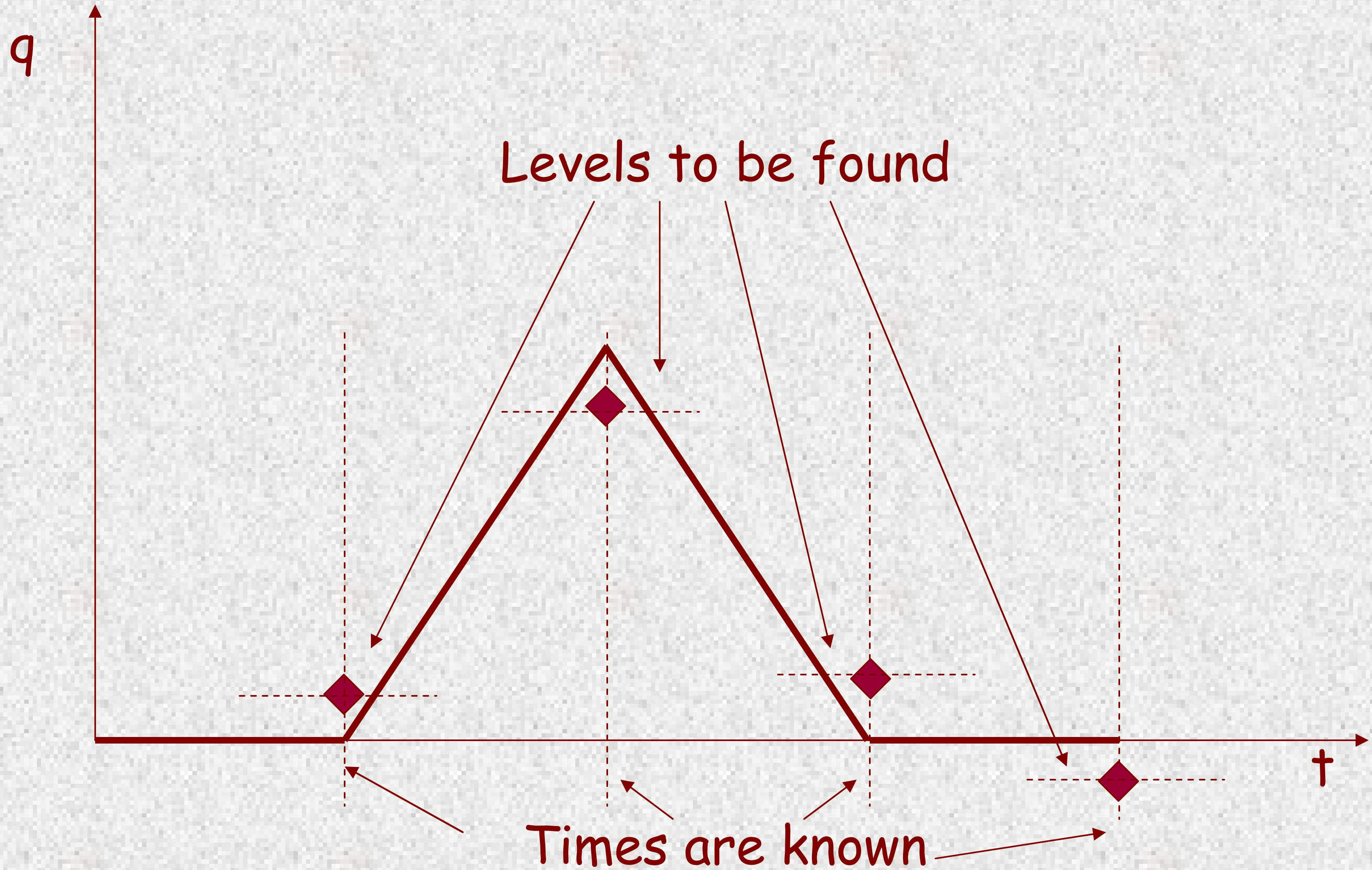
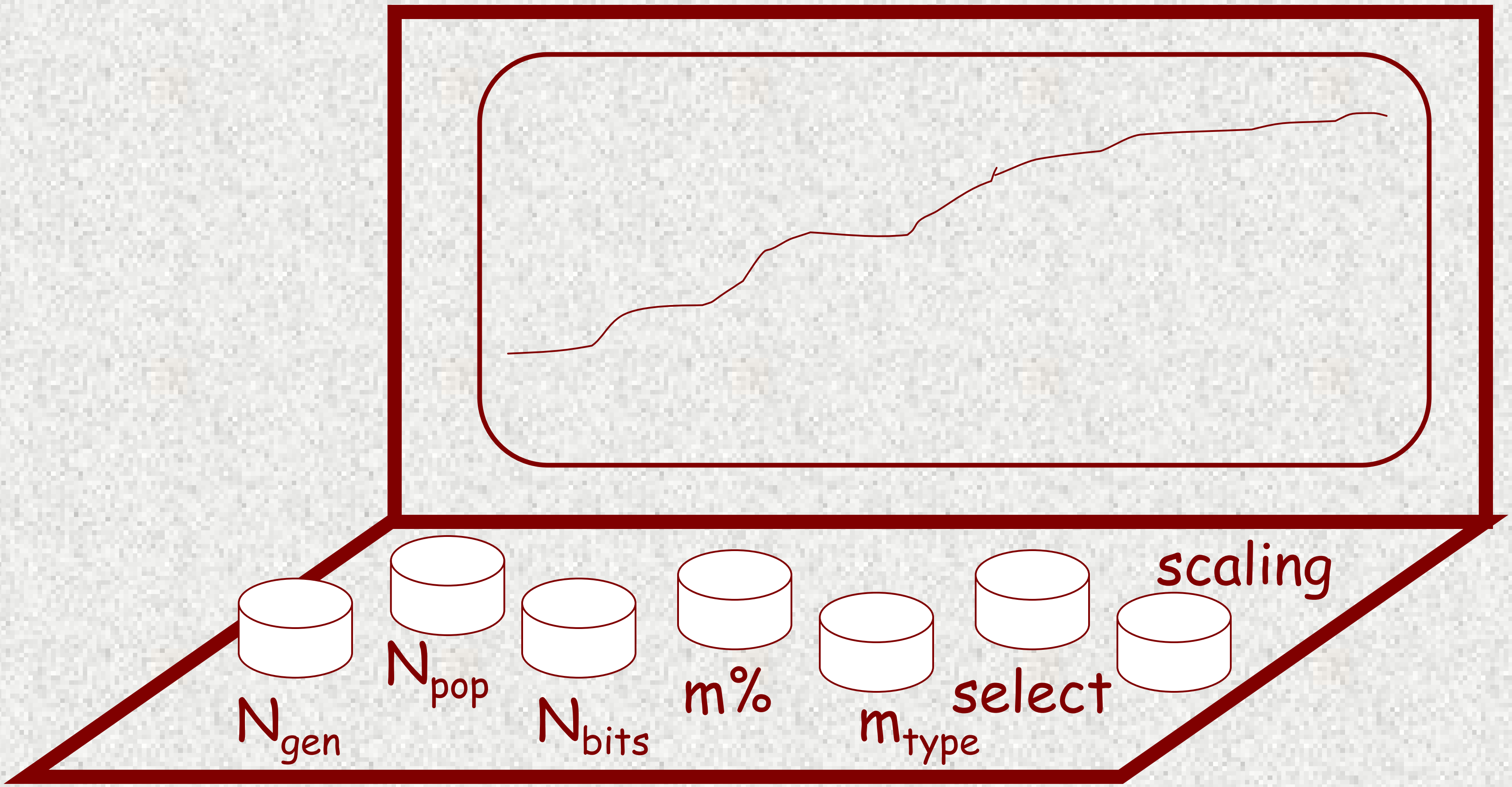


FIGURE 5.3 Triangular heat flux for test case. Finite insulated plate.

Function Parameterization



GA "Machine"



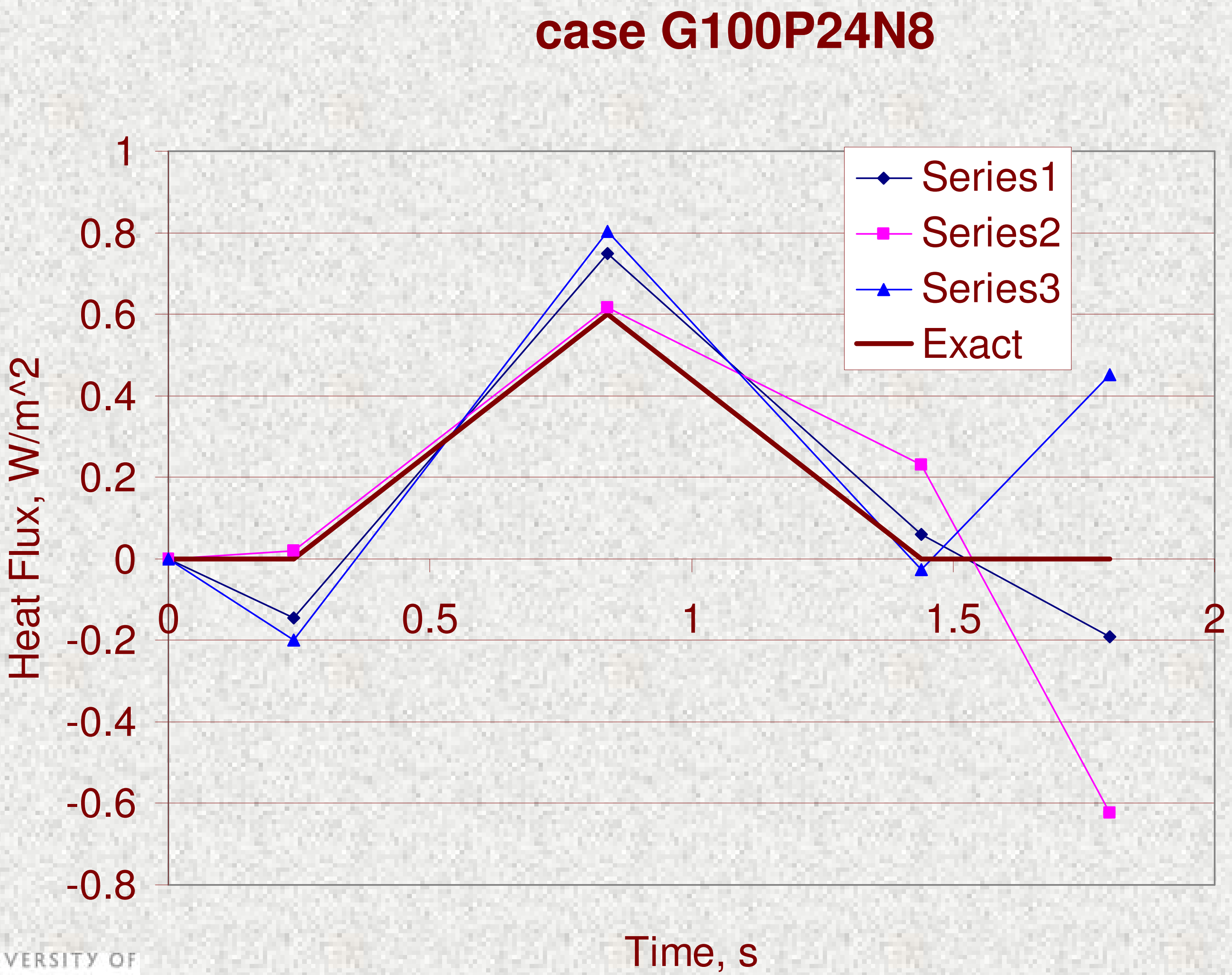
Of course…..

- Want the best solution in the shortest number of generations

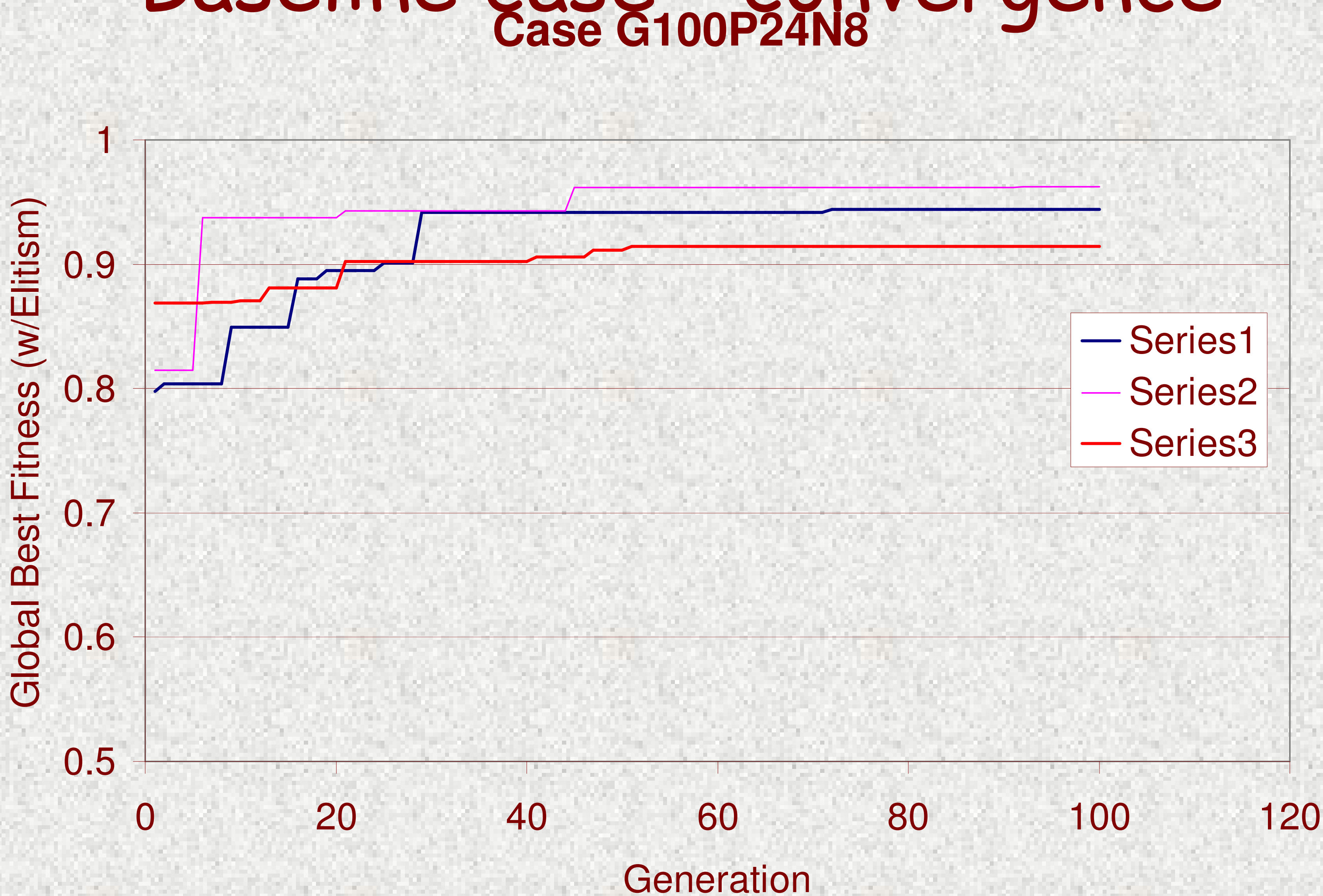
Baseline Configuration

- N_{gen} = 100
- N_{pop} = 24
- N_{bits} = 8 (resolution 0.008)
- 10% mutation
- Bitwise mutation
- Roulette Wheel Selection

Baseline case - results



Baseline case - convergence



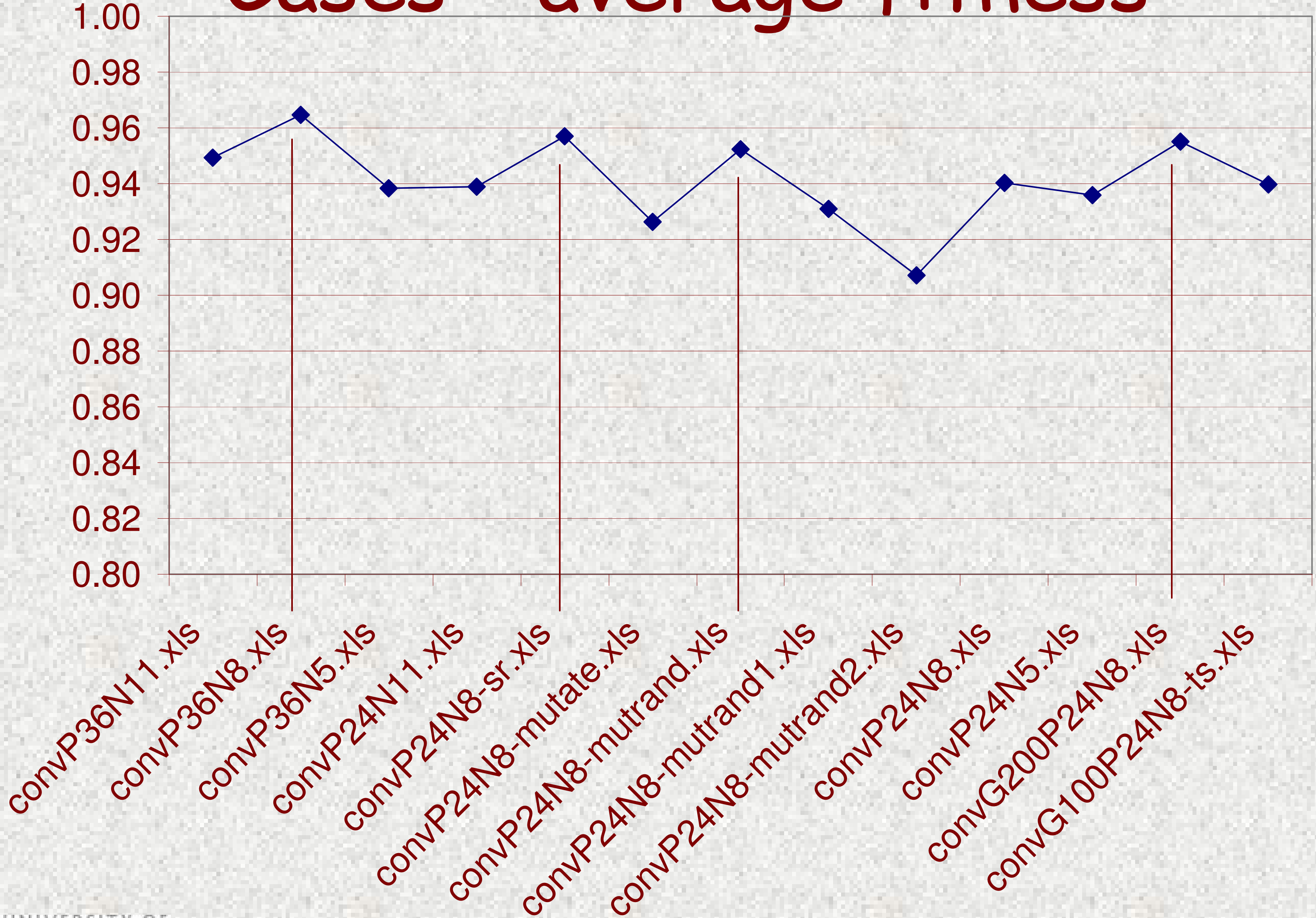
Basic idea to select parameters

- Turn one knob at a time
- Run GA three times for each
- Compare results (fitness function)
- Look at convergence histories

Cases considered

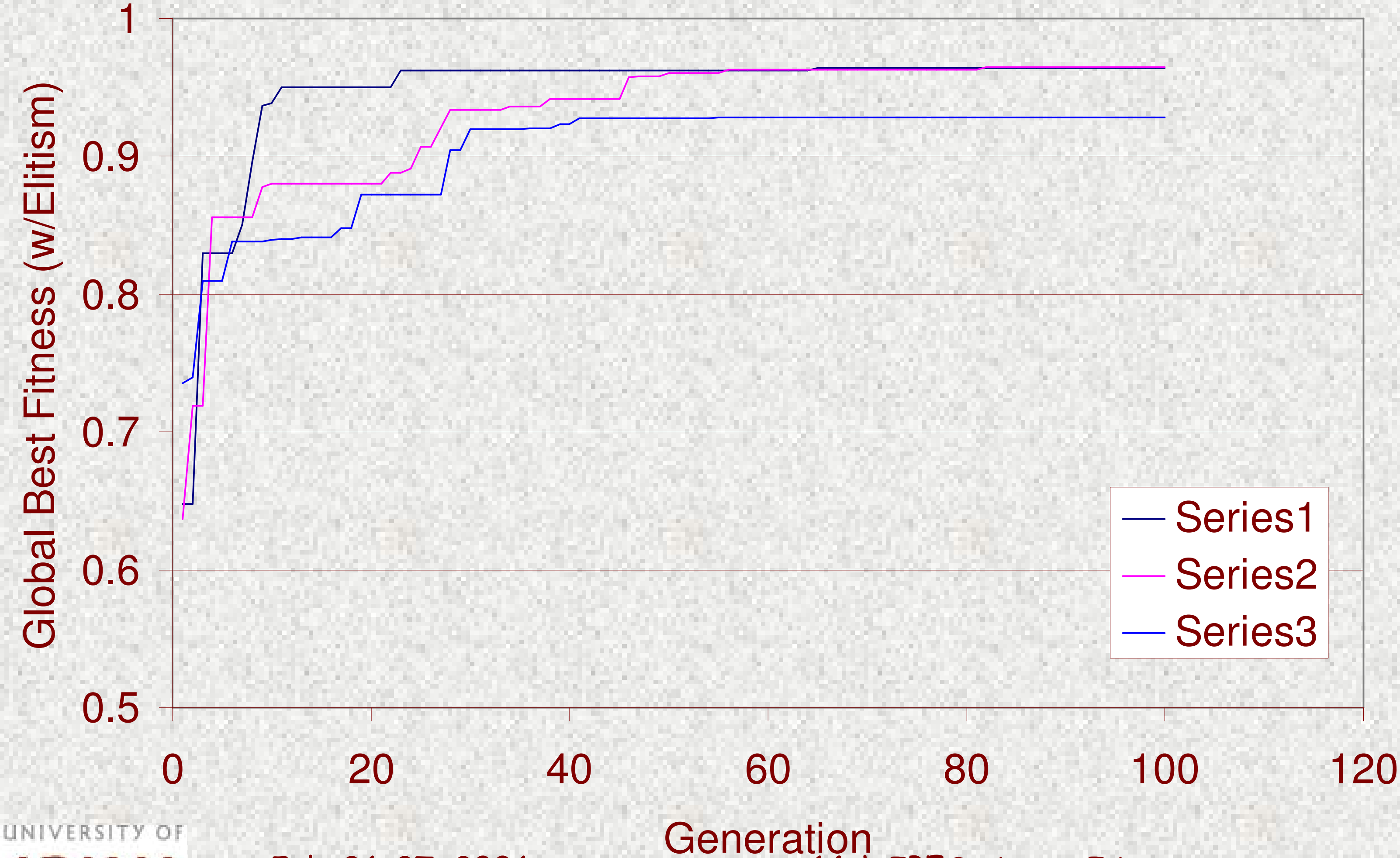
name	gen	pop	nbits	mtype	m%	select	scaling
convP36N11.xls	100	36	11	bit inversion	10%	roulette wheel	none
convP36N8.xls	100	36	8	bit inversion	10%	roulette wheel	none
convP36N5.xls	100	36	5	bit inversion	10%	roulette wheel	none
convP24N11.xls	100	24	11	bit inversion	10%	roulette wheel	none
convP24N8-sr.xls	100	24	8	bit inversion	10%	stochastic remainder	none
convP24N8-mutate.xls	100	24	8	bit inversion	25%	stochastic remainder	none
convP24N8-mutrand.xls	100	24	8	bit randomization	10%	stochastic remainder	none
convP24N8-mutrand1.xls	100	24	8	bit randomization	25%	stochastic remainder	none
convP24N8-mutrand2.xls	100	24	8	bit randomization	1%	stochastic remainder	none
convP24N8.xls	100	24	8	bit inversion	10%	roulette wheel	none
convP24N5.xls	100	24	5	bit inversion	10%	roulette wheel	none
convG200P24N8.xls	200	24	8	bit inversion	10%	roulette wheel	none
convG100P24N8-ts.xls	100	24	8	bit inversion	10%	tournament	none
convG100P24N8-scale.xls	100	24	8	bit inversion	10%	roulette wheel	linear

Cases – average fitness



Convergence - Bit Randomization

Case G100P24N8-mr10%



Observations

- SGA has been used to obtain solutions to an IHCP using a commercial solver in a parallel computer
- Stochastic remainder for selection and bitwise randomization for mutation appear to yield better solutions
- GA solutions are qualitatively good but not excellent

Observations