Lecture L2: Evolutionary Optimization

- Computational intelligence
- Evolutionary algorithms (EAs)
- Binary-coded genetic algorithms (GAs)
- Advantages and disadvantages
- Some example case studies on solving optimization problems

Evolutionary Optimization:
A Motivation from Nature

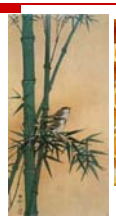
- Natural evolution + genetics
  - Guided search procedure
  - Offspring are created by duplication, mutation, crossover etc.
  - Good solutions are retained and bad are deleted
  - Information is coded





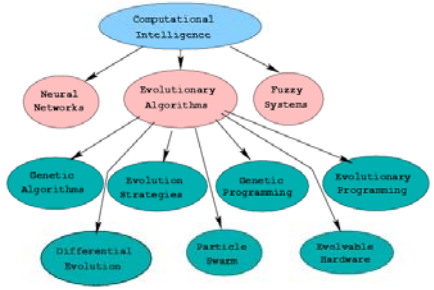
Nature as an Optimizer

- Nature as structural engineer
  - Stem, Bamboo, insect trachea, bee-hive
- Nature as a CFD solver
  - Birds, fishes
- Nature as a drag reducer
  - Penguin body






Computational Intelligence and Evolutionary Algorithms (EAs)

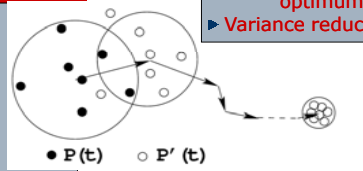


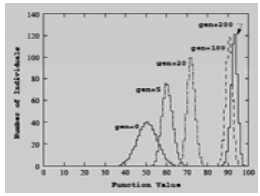
- We treat an EA here as a search and optimization tool

Evolutionary Algorithms as Optimizers

- Mean approaches optimum
- Variance reduces

```
begin
Solution Representation
t := 0;  // generation counter
Initialize P(t);
Evaluate P(t);
while not Terminate
do
   P'(t)  := Selection (P(t));
   P''(t) := Variation (P'(t));
   Evaluate P''(t);
   P(t+1):= Survivor (P(t),P''(t));
   t := t+1;
od
end
```

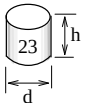




Traditional Evolutionary Approach: *Binary-Coded Genetic Algorithms*

- Genetic Algorithms (John Holland, 1962)
- Design of a can for minimum cost having at most *V*
- volume
- Objective function: Cost(*d,h*): $f(d,h) = \pi dh + 2\pi d^2/4$
- Constraint: Volume (*d,h*): $\Pi d^2 h/4 \geq V$
- Representation in binary strings
- Fitness: objective value + penalty for constraint

Solution Representation:



(d, h) = (8, 10) cm
(Chromosome) = 0 1 0 0 0 0 1 0 1 0

Mapping Binary Strings to Real Numbers

- *x* in [*a,b*] represented by an *l*-bit substring

$$x = a + \frac{b-a}{2^l - 1} DV(s)$$

- *DV(s)* is the decoded value of substring *s*
- 00000 maps to *a*
- 11111 maps to *b*
- Any other substring maps to a number within *a* and *b* (fixed and discrete coding)
- Get more precision with more *l*

An Example of Binary Coding

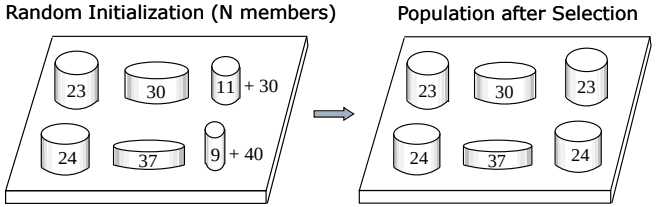
- Say x in [2,10], l=5
- x=2 + 8/31 DV(s)
- Say, s = (01011), DV(s) = 11
- x=2 + 8/31 (11) = 4.8387
- Precision = 8/31 = 0.258
- Want precision = 0.01 = $8/(2^l-1)$
- $l = \log_2 (1+8/0.01) = 9.65$ or 10
- Other nonlinear scaling possible

Genetic Algorithm: A Hand Simulation

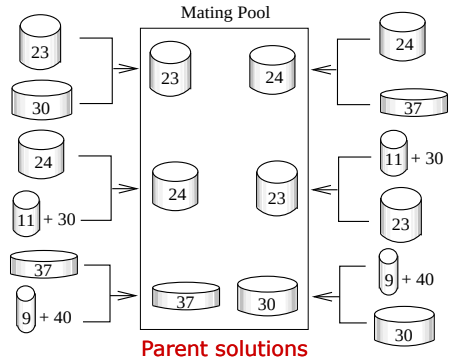
Initialization and Evaluation:

- Fitness = Cost + Penalty (proportional to constraint violation)

Selection: *Emphasizing better solutions*



Tournament Selection Operator

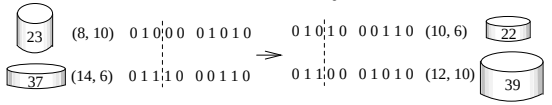


Tournament Selection Alone

- Repeated application will increase more and more copies of population best solution
  - Not particularly interesting
  - Better computing algorithms available
- Population diversity reduces with iteration
  - Has converging properties

Variation Operators:
Creating new offspring solutions

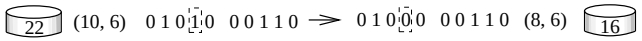
- Crossover operator with p_c:



- Crossover operator alone, repeated application
  - Say, strings 11111 11111 and 00000 00000
  - Eventually create a randomly assorted population
  - Not an interesting operation alone
  - Population diversity increases with iteration

Variation Operators:
Mutation Operator

- Mutation operator with p_m:

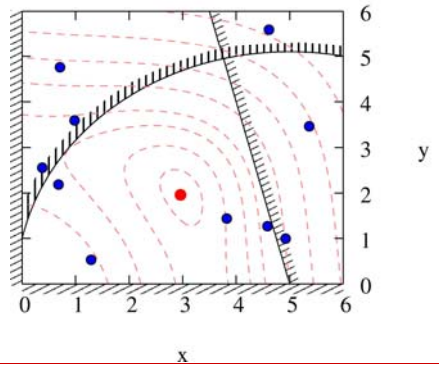


- Mutation operator alone, repeated application
  - Say on string 00000 00000 with pm
  - Eventually create a randomly assorted population
  - Not particularly interesting operator alone
  - Population diversity increases with iteration

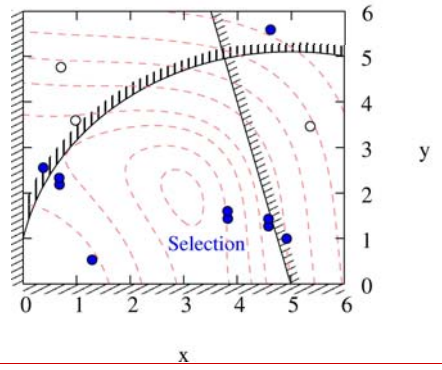
Combination of GA Operators Meaningful

- Innovation properties
  - Selection + Recombination
  - Good sub-solutions recombines to create good larger sub-solutions
  - An ingredient for creating innovation
- Intelligent Hill-climbers
  - Selection + Mutation
- Reliable optimizers with selection+recombination+mutation

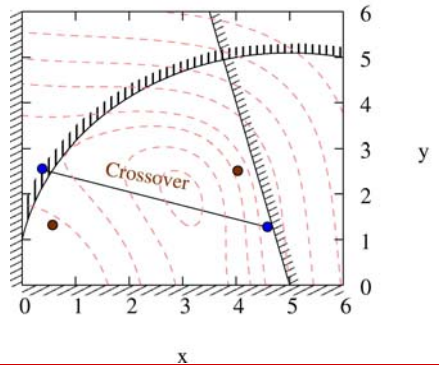
An Illustration of GA's Working Principle: Initial Population



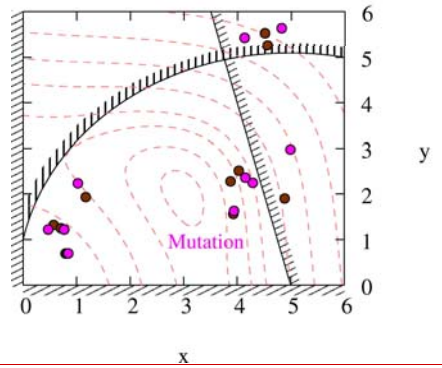
Selection Operation



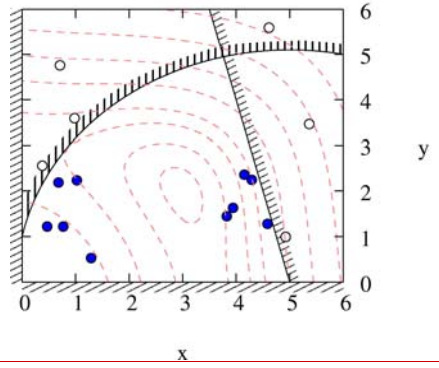
Crossover Operation



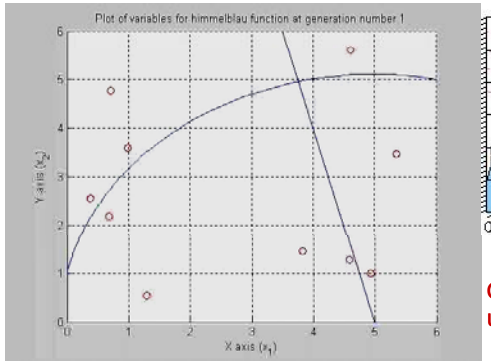
Mutation Operation

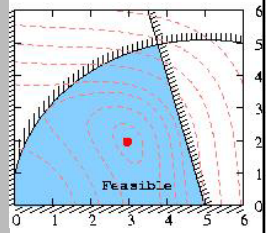


New Population (After Elite Preservation)



Simulation of a GA





Global search ended up with hill-climbing

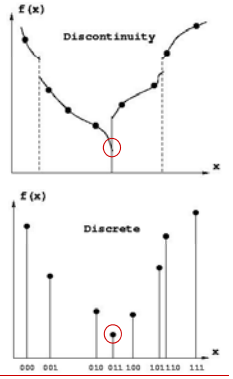
Differences with Classical Methods

- GAs work on a coding of variables
  - Mixed variable types can be handled
  - Permutation, continuous, discrete together
- GAs are population-based
  - Safety in numbers -> global perspective
  - *Implicit parallelism*
- GA operators are probabilistic
  - Less chance of getting stuck
- GAs do not use any gradient information
  - Black-box search -> wide applicability
- Ideal for parallel computation
  - Distribute evaluations among processors and more

Non-smooth Problems

Handling discrete, discontinuous variables

- Non-differentiability, discontinuity, discreteness, non-linearity not a difficulty
- GAs work with a discrete search space anyway
  - So, GAs are natural choice for discrete problem solving



Evolution of Evolutionary Algorithms

- GAs: John Holland in 1962 (UMich)
- ESs: Rechenberg and Schwefel in 1965 (Berlin)
- EP: Larry Fogel in 1965 (California)
- First ICGA: 1985 in CMU
- First GA book: Goldberg (1989)
- First FOGA workshop: 1990 in Indiana (Theory)
- First Fusion: 1990s (*Evolutionary Algorithms*)
- Journals: ECJ (MIT Press), IEEE TEC, GPEM (Kluwer), Natural Computation (Elsevier)
- GECCO and CEC since 1999, PPSN since 1990
- About 20 major conferences each year
- ISGEC merged with ACM since 2005 as SIGEVO

Case Study 1: Tracking Suspect

- Caldwell and Johnston, 1991
- Trend: Artist interacting with people
- Recognition in oppose to recall
- Difficult to use a classical method
  - Discrete representation
- Forehead, eyes, nose, mouth and chin (7-bits each)
- Fitness: One on a nine-pt. scale
- Population size=20, SUS selection, 1-pt crossover and bit-wise mutation

Simulation Results (Tracking Suspect)

- Original (left) and Simulated (right)



- Idea can be used in non-mathematical problem solving (aesthetics)

Some Features of an Evolutionary Optimization

- Must compete against contemporary algorithms: Quasi-Newton, SQP, Trust-region methods etc.
- Fundamental differences:
  - Population approach: How many members?
  - Stochastic search principles: Argument for global optimization
  - Direct search approach: More widely applicable
- Generic EA operators are too naive and simple
- Most EA applications use knowledge-augmented operators or a hybrid approach

Advantages of EAs

- Applicable in problems where no (good) method is available
  - Discontinuities, non-linear constraints, multi-modalities
  - Discrete variable space
  - implicitly defined models (*if-then-else*)
  - Noisy problems
- Most suitable in problems where multiple solutions are sought
  - Multi-modal optimization problems
  - Multi-objective optimization problems
- Parallel implementation is easier

Disadvantages of EAs

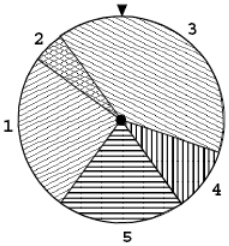
- No guarantee for finding optimal solutions in a finite amount of time
  - However, asymptotic convergence proofs are available
  - For specific problems, computational complexity worked out
- Parameter tuning mostly by trial-and-error: Self-adaptation is a remedy
- Population approach may be expensive: Parallel implementation is a remedy

Fitness Assignment Schemes

- From objective/constraint functions
  - Usual practice
- From subjective considerations
  - Population-specific (more like ranking)
- From ensemble of population members
  - Survival of population dictates
- From simulation
  - Concept evaluation

Other Selection Operators

- Roulette Wheel Selection
  - Proportionate with non-negative fitness
  - Scaling problems
  - Noisy, SUS is a compromise
- Ranking Selection
- Truncation Selection



Other Binary Crossover Operators

- Random yet structured information exchange
- Single-point crossover:
  1 1 1|1 1 1 1 → 1 1 1|0 0 0 0
  0 0 0|0 0 0 0 → 0 0 0|1 1 1 1
- Two-point Crossover
  1 1 1|1 1|1 1 → 1 1 1|0 0|1 1
  0 0 0|0 0|0 0 → 0 0 0|1 1|0 0
- Uniform crossover
  1 1 1 1 1 1 1 → 1 0 1 0 0 1 0
  0 0 0 0 0 0 0 → 0 1 0 1 1 0 1

Less structure ↓ More search

Further Crossover Operators

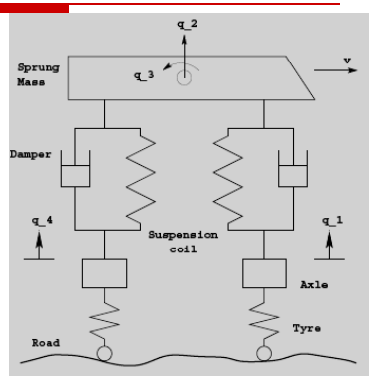
- Reduced Surrogates (Booker, 1987)
  - Not all crossings are productive
    010110100 → 1011
    001000100 → 0100
    (Reduced surrogate)
- Choose cross sites among reduced surrogates, map back to original strings
- Marginal gain in performance

Other Mutation Operators

- Phenotypic mutation
  - Avoid Hamming cliff: d(01111,10000)=5
  - Take unit step in decoded space
  - Overcome with Gray codes
- Adaptive mutation
  - Use a meta-GA
  - Mutation-controlling genes
  - Temperature related $p_m=exp(-\theta/T)$ (Sirag and Weisser, 1987)
  - Diversity controlled mutation (Callaway, 1991)
    - More diversity, less mutation

Case Study 2: A Car Suspension Design

- Suspension isolates passengers from road excitations
- Dampers and suspension coils are design variables
- Minimize Transmissibility or Maximum Ride comfort (ISO 2631)
- Constraints on maximum jerk, natural frequencies, etc.



Mathematical Model

- Four degrees of freedom

$$\ddot{q}_1 = (F_2 + F_3 - F_1)/m_{fu}, \quad \ddot{q}_2 = -(F_2 + F_3 + F_4 + F_5)/m_s,$$
$$\ddot{q}_3 = [(F_4 + F_5)\ell_2 - (F_2 + F_3)\ell_1]\,/J, \quad \ddot{q}_4 = (F_4 + F_5 - F_6)/m_{ru},$$

- Forces in each spring and damper:

$$F_1 = k_{ft}d_1, \quad F_2 = k_{fs}d_2, \quad F_3 = \alpha_f \dot{d}_2,$$
$$F_4 = k_{rs}d_4, \quad F_5 = \alpha_r \dot{d}_4, \quad F_6 = k_{rt}d_3.$$

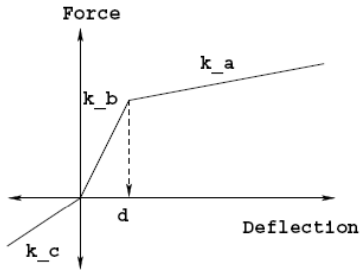
- Deflections of each mass:

$$d_1 = q_1 - f_1(t), \quad d_2 = q_2 + \ell_1 q_3 - q_1,$$
$$d_3 = q_4 - f_2(t), \quad d_4 = q_2 - \ell_2 q_3 - q_4.$$

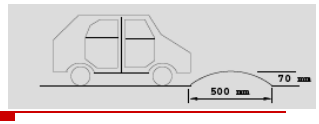
- Use numerical integration to find responses

Practical Considerations

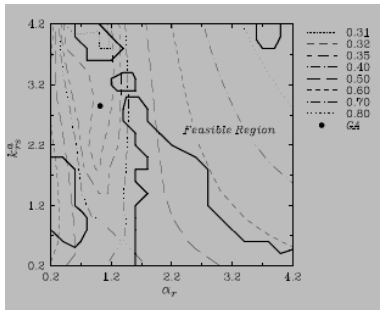
- Nonlinear variation of spring rate and damper characteristics
- Front natural freq. less than rear natural freq.
- Maximum vertical jerk is limited for comfort



Proof-of-Principle Results



- Over a sinusoidal bump
- m_s=730 kg, m_{fu}=50 kg, m_{ru}=115 kg, L=2.85 m,
- v=5 km/hr, J=2.89(10^4) kg.m^2
- Jerk is only constraint
- Two variables (k_{fs} and α_f) fixed
- Optimum is surrounded by infeasible islands



Classical Methods First: MATLAB Solution

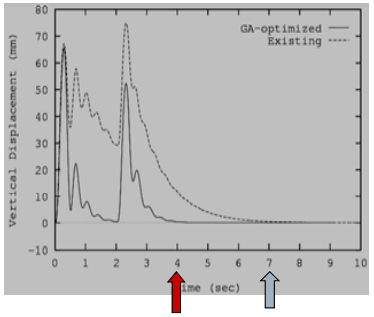
- Sequential Quadratic Programming (SQP) from Optimization Toolbox
- Most cases hang saying
  - *Matrix is near singular or badly scaled, results may be inaccurate*
- Parameter change requires 7 hours on Pentium (133 MHz)
- Solution depends on initial guess

$$(1.00,1.00,1.00,1.00) \quad 0.96 \quad \rightarrow \quad (4.31,0.23,0.83,4.99) \quad 0.49$$
$$(1.00,1.00,1.50,0.90) \quad 0.89 \quad \rightarrow \quad (3.76,2.89,3.10,2.19) \quad 0.48$$

Binary GA Solution

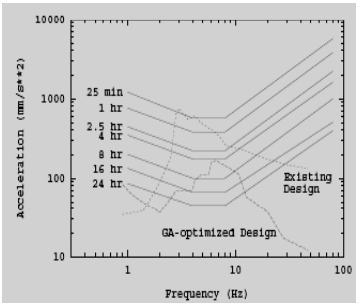
- Binary-coded GA (pop size = 30, string length = 30, p_c=0.8, p_m=0.01)
- Constraints handled by bracket-operator penalty function
- GA Solution: k_rs=2.86, α_r=1.01 Trans. = 0.315
- Existing solution: k_rs=1.45, α_r=1.00 Trans. = 0.82
- GA solution 64% better (Jerk = 5.1 < 18 m/s^3)

Response of Optimized Solution

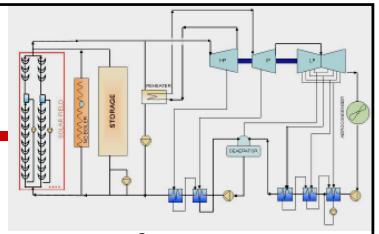


Comfort as per ISO-2631

- Car over a random road
- Maximum acceleration versus frequency of motion
- Minimize area under accl.-freq. curve with constraints
- GA solution 240 min comfortable (versus existing 25 min)



Case Study 3: Solar Thermal Plant Design



- Three variables:
  - Solar collector area

    A_C (m^2), less than 750,000 m^2 for practicalities
  - Storage capacity, E (kJ)
    - One tank has 8 hours of storage capacity
    - More storage needs a jump in cost due to fixed cost in building more tanks
  - Auxiliary Boiler Power, P_{AUX} (kW)

Basic Operating Principles

- Evaluate direct solar radiation and estimate steam mass flow rate
- If ≥75% load can be supplied, use solar energy
- If >100% load can be supplied, store the remaining energy
- If <75% cannot be supplied, used stored energy
- If not enough in storage and solar radiation, use auxiliary boiler source (max of 15% of total as a regulation)

Optimization Problem

$$\begin{aligned} \text{maximize} \quad & P(A_C, E, P_{AUX}), \\ \text{subject to} \quad & 0 \le A_C \le 750000, \\ & 0 \le E, \\ & 0 \le P_{AUX}, \end{aligned}$$

- Profit, *P=I-C*
- Non-mathematical, black-box type
- Many physical constraints, involving if-then-else statements
- Operating principles as constraints

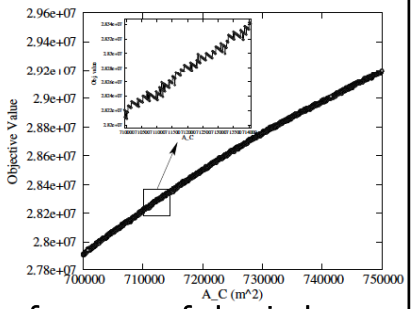
Classical Methods First

- LGO software
  - *A global branch and bound solver, a global adaptive random search solver, a multi-start global random search solver, a constrained local search solver based on the reduced gradient*
- LINDO software
  - *Simplex based solvers for LP, an interior point solver for linear and quadratic programming problems, mixed-integer programming solvers*
- Negative profit
- Initial soln. does not change much, stuck at subopt. soln.

Initial Solution				LGO			
$P(\mathbf{x})$	A_C	E	P_{AUX}	$P(\mathbf{x})$	A_C	E	P_{AUX}
27049676	693450	5405000000	174300	22068500	862768	6720680000	0
23514216	594975	2492000000	268100	22068500	862768	6720680000	0
29172500	749000	6350000000	92800	22068500	862768	6720680000	0

Investigating Nature of Search Space

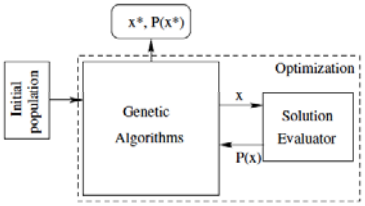
- Five sets of 10,000 reported
  - No attractor
- Investigate search region
- E = 6.34(10^9) kJ
  P_{AUX} = 92768.3 kW
  A_C in [700,000, 75
- Massive multi-mod
- Explains the poor performance of classical methods



Genetic Algorithms

- First, a standard real-parameter GA
  - Tournament selection, SBX, polynomial mutation
- E ≤10^{20}, P_{AUX} ≤10^{20} assumed
- Evaluator as a black-box
- Popsize=50, maxgen=150

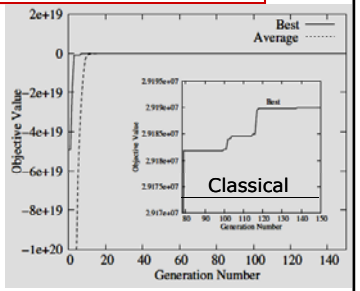
rga.c code from http://www.iitk.ac.in/kangal/soft.htm



GA Results

$A_C = 749,980.86$ m^2, $E = 6,191,823,943.05$ kJ,
$P_{AUX} = 92,898.24$ kW, $P(\mathbf{x}) = 29,189,994.89$ €.

- Profit better than before
- A_C close to upper bound, increase in upper bound will allow better profit
- Significant improvement from initial population



Multiple GA Simulations

- Quite consistent results with different initial populations
- Different orders of magnitude in variables

$P(\mathbf{x}^*)$ (€)	$\mathbf{x}^*$		
	A_C (m^2)	E (kJ)	P_{AUX} (kW)
29195554.85	749996.42	6323010005.53	92816.46
29195243.43	749930.47	6384646915.43	92807.45
29193008.27	749970.19	6346284954.49	92826.02
29192668.30	749998.00	6339863781.14	93364.93
29190841.29	749963.98	6171262677.77	92931.44
29189994.89	749980.85	6191823943.05	92898.24
29188480.61	749807.35	6410135972.39	93238.03
29188027.91	749991.77	6339257051.97	92921.15
29183821.73	749872.41	6209282091.91	93061.24
29163020.97	749749.25	6191824038.55	95266.52

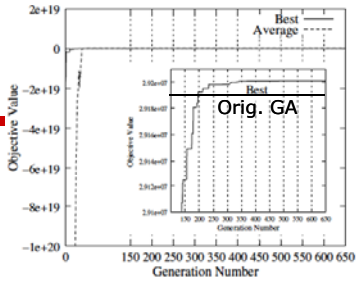
Improved GA

- After first 150 gen, note x* and pop std dev (σ) of each variable
  - x in [x*-σ, x*+σ]
- Run GA for next 50 generations
  - Keep all previous solutions within the bounds, add random solutions to fill population
- If best does not improve, terminate
- Better result (0.04%) is obtained

$$A_C = 749,999.99 \text{ m}^2, \quad E = 6,346,926,947.98 \text{ kJ},$$
$$P_{AUX} = 92,768.27 \text{ kW}, \quad P(\mathbf{x}) = 29,201,019.61 \text{ €}.$$

Improved GA (cont.)

- Gradual improvement
- Multiple runs are consistent



$P(\mathbf{x}^*)$ (€)	$\mathbf{x}^*$		
	A_C (m²)	E (kJ)	P_{AUX} (kW)
29201019.61	749999.99	6346926947.98	92768.27
29201018.75	749999.94	6346929408.75	92768.26
29201018.62	749999.94	6346929669.93	92768.26
29201018.62	749999.94	6346929478.79	92768.27
29200967.67	749997.61	6347055515.19	92768.46
29200967.42	749997.62	6347055515.19	92768.46
29200957.23	749997.01	6347087739.59	92768.27
29200953.07	749997.00	6347092303.42	92768.31
29200922.51	749999.96	6346929711.66	92777.49
29200522.97	749992.62	6318673893.16	92768.70

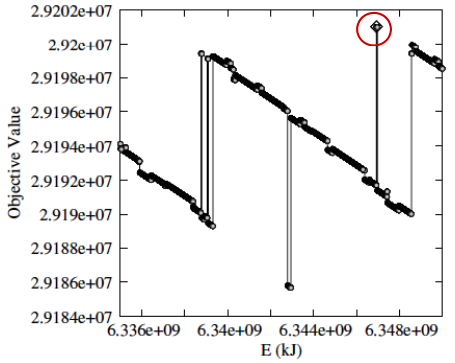
- Profit is 5,465 Euros better than original GA
- 0.0017% difference in profit

Practical Interpretation

- Radiator capacity (A_C) at its upper limit
- P_{AUX} value is eqv. to 75% of power production with auxiliary boiler alone
- Total storage (E) is one full tank and another tank with 90% capacity
- Hybrid system with exactly 15% from boiler -> satisfies regulatory limit
  - Active constraint due to cheaper power production form boiler

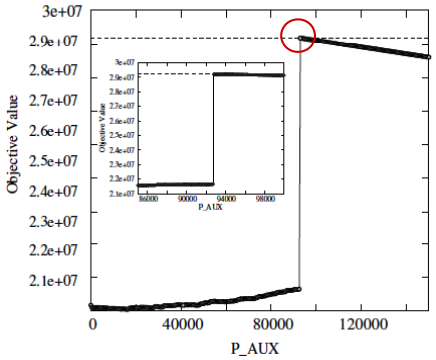
Analysis of Obtained Solution

- Keep A_C and P_{AUX} fixed, vary E
- Optimum is isolated
- Improved GA is able to find it



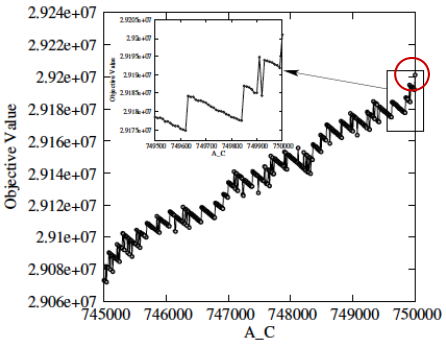
Analysis of Obtained Solution (cont.)

- Keep A_C and E fixed, vary P_{AUX}
- Optimum is at a discontinous point
- Improved GA is able to find it



Analysis of Obtained Solution (cont.)

- Keep P_{AUX} and E fixed, vary A_C
- Optimum is at a variable boundary
- Search space is multimodal
- Improved GA is able to find it



Summary of Solar Power Plant Optimization

- Despite having three variables
  - *f* and constraint computations are in a black-box
  - *f* has massive multi-modalities
  - *f* has discontinuities
  - ***x**** is a discontinuous point
  - ***x**** lies on a variable boundary
  - ***x**** is isolated
  - ***x**** variables are of different (five) orders of magnitude
- All above provided difficulties to many classical methods
- Continuously focused GA found robust

Popular Binary GAs

- CHC (Eshelman, 1992)
  - Selection: Truncation selection from P_tUQ_t
  - Incest Prevention (two strings with minimum 25% bit diff. are mated)
  - HUX: Exchange 50% of different bits
  - Allows restarts when stuck
- Genitor (Whitley, 1989)
  - Tournament selection, reduced surrogate based crossover, best of offspring replace worst
  - Steady-state GA

Conclusions

- Still a need for a robust optimizer
- *Evolutionary algorithms* are good candidates for difficult problems
- EAs differ from classical methods
  - Population-based
  - Stochastic operators
  - No use gradients, in general
  - Flexible and can be widely used

