Application examples of EAs

- Nuclear reactor core reload design
- Set partitioning - airline crew scheduling
- 2-dimensional packing
- Hot water / steam turbulent nozzle
- Truss layout
- Concrete shell roof
- Dipole magnet structures
- Optical multilayer design
- Remarks

Nuclear reactor core reload design (1)

Goal: Find a cost-minimizing arrangement of fuel assemblies.

- Refueling with fresh assemblies.
- Change of position and orientation of partially burned assemblies.

Combinatorial search space size:

$$\frac{69!}{(69-48)!} \cdot 48! \cdot 4^{48} \approx 1.0 \cdot 10^{107}$$

Nuclear reactor core reload design (2)

Structure of individuals:

Quarter core	9	10	11	12	13	14	15
H	A552	A493	0121	A492	A579	0125	A180
G	0129	A491	A489	0133	A404	0101	A174
F	A485	0141	A407	A550	A408	0145	A177
E	A487	A396	0153	A554	0157	0109	A178
D	0137	A548	A546	A483	0165	A173	
C	A397	A385	0161	0169	A588	A179	
B	0105	0149	0113	A171	A155		
A	A167	A163	A159				

Stock		
0117	A204	A392
0173	A206	A393
0177	A229	A398
A156	A231	A399
A160	A386	A403
A164	A387	A409
A168	A388	A577

Representation

	H9	H10	H11		A10	A11	1		20	21
Fuel assemblies	17 (A552)	4 (A493)	67 (0121)		53 (A163)	22 (A159)	11 (0117)		45 (A388)	3 (A577)
Orientations	-3	0	-2		-1	-3	-3		-2	0

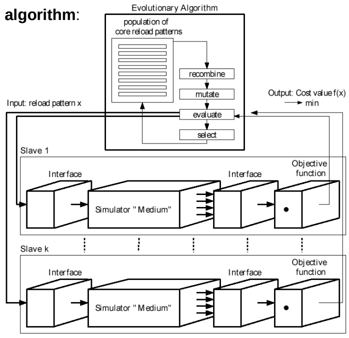
Quarter core | Stock

48 Fuel assemblies + Orientations

Simulator-Input

Nuclear reactor core reload design (3)

Parallel evolutionary algorithm:



Master-Slave concept

Nuclear reactor core reload design (4)

Simulator requirements:

- 50 MB disk space per evaluation.
- 28 MB main memory.
- 10 min. CPU-time (SPARC 10/40)

Nuclear reactor core reload design (5)

Mutation:

- Exchange of 2 randomly chosen fuel assemblies
  - Quarter core ↔ quarter core or
  - Quarter core ↔ stock;
- Variation of orientations
  - uniformly distributed over ({0, -1, -2, -3} - actual orientation),
  - application probability $p \approx 0.05$.
- Symmetry-preserving mutation
  (assumption: 1/8-core symmetry)

Heuristics:

- No fresh fuel assemblies at the core's border.
- Gradient of border elements directed towards the inner core.

Nuclear reactor core reload design (6)

Objective function:

$$\boxed{f = \frac{205}{a_1} + \frac{a_2}{40} + \frac{a_3}{2.8} + \frac{a_4}{2.8} + \frac{\max\{a_3, a_4\}}{1.4}}$$

a_1, … , a_4: Results from the simulator run.

- a_1: Cycle length (number of full load days).
- a_2: Number of fresh fuel assemblies.
- a_3: Rel. avrg. FA-performance after 6 full load days.
- a_4: Rel. avrg. FA-performance when the relative maximum is reached.

Nuclear reactor core reload design (7)

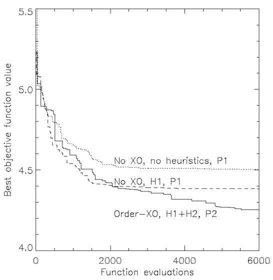
Constraint: Cycle length: $\underline{T} \leq a_1 \leq \overline{T}$

Incorporation by time-varying penalty term:

$$p(a_1) = \begin{cases} (a_1 - \overline{T}) \cdot i/c, & a_1 > \overline{T} \\ (\underline{T} - a_1) \cdot i/c, & a_1 < \underline{T} \end{cases}$$

- i: Number of evaluations so far
- c: Constant ($c = i_{max}$).

Nuclear reactor core reload design (8)



- Representative runs.
- Crossover, heuristics, dynamic penalty term: clear advantages.
- 1 day of computing time!
- Hand optimized expert solution: 4.16
- Best EA-solution: 4.09

Set partitioning problem (1)

Problem instance:

- Vector $\overline{C} = (c_1, \ldots, c_n)$ of costs.
- $m \times n$ matrix A, $a_{ij} \in \{0, 1\}$.

Feasible solution:

Vector $\overline{x} \in \{0, 1\}^n$ such that $\forall i \in \{1, \ldots, m\}$

$$\sum_{j=1}^{n} a_{ij} \cdot x_j = 1$$

Objective function:

$$\sum_{j=1}^{n} c_j \cdot x_j \to \min$$

Set partitioning problem (3)

Example:

Legs	Pairings, $m = 9$, $n = 6$					
HH → F	1	1	0	0	1	0
F → M	0	1	0	1	0	0
F → B	0	0	1	0	1	0
B → M	0	0	1	0	1	0
M → HH	0	1	0	0	1	0
F → HH	1	0	0	0	0	1
M → F	0	0	1	1	0	0
B → F	0	0	0	0	0	1
HH → B	0	0	0	0	0	1
Costs	250	100	80	140	60	200

$$\sum_{j=1}^{n} a_{ij} \cdot x_j = 1$$

$$f(x) = \sum_{j=1}^{n} c_j \cdot x_j$$

Some solutions:

- Feasible: $\overline{x}_1 = (0, 1, 1, 0, 0, 1)$, $f(\overline{x}_1) = 380$.
- Feasible: $\overline{x}_2 = (0, 0, 0, 1, 1, 1)$, $f(\overline{x}_2) = 400$.
- Infeasible: $\overline{x}_3 = (0, 0, 1, 1, 0, 0)$, fails on rows (legs) 1, 5, 6, 8, 9.

Set partitioning problem (2)

Airline crew scheduling:

- Rows of A: *flight legs* (nonstop flights between city pairs).
- Columns of A: *pairings* (possible crew round trips).
- Costs c_j of pairing j may depend on: flight times, hotel costs, deadheads, waiting times, contracts, etc…

Set partitioning problem (4)

Objective function:

$$f(\bar{x}) = \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{m} k_i \left(\sum_{j=1}^{n} a_{ij} x_j - 1 \right)^2 \rightarrow \min$$

- k_i are weighting factors
- if the condition

$$\sum_{j=1}^{n} a_{ij} \cdot x_j = 1$$

is violated:

- add *penalty term*:

$$k_i \cdot \left(\sum a_{ij} x_j - 1 \right)^2$$

Set partitioning problem (5)

Levine (1993) compared some strategies (m = 23, n = 3068):

- standard GA
- Traditional local search "row-heuristic"
- Traditional local search "unit-heuristic"
- Hybrid GA + local search "row-heuristic"
- Hybrid GA + local search "unit-heuristic"

Results:

	# runs	# feas.	# opt.	f^*
GA	50	0	0	—
GA + "row"	47	47	5	6678
GA + "unit"	50	50	0	8294
"row"	50	50	0	7642
"unit"	50	0	0	—

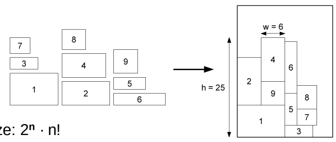
2-dimensional packing (1)

- Placement of advertisements on newspapers.
- Container- and truck-packing.
- Module placement in chip design.

2-dimensional packing:

Given rectangular forms $Q_1, \ldots, Q_n$ and a rectangular packing form, find a packing that maximizes total height.

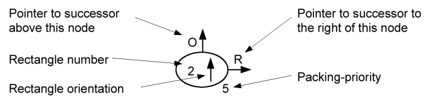
Example:



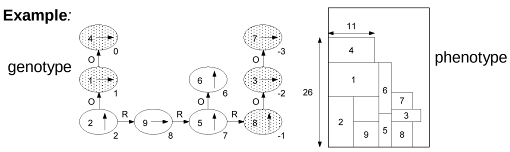
Search space size: $2^n \cdot n!$

2-dimensional packing (2)

Representation of packing pattern: Binary tree.



Example:



BL-rule: No rectangle can be shifted to the left or bottom without creating an overlapping.

2-dimensional packing (3)

Mutation:

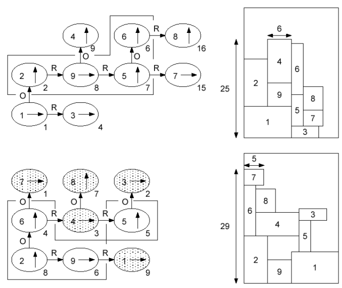
- Change a node priority.
- Change an orientation.
- Exchange node positions within the tree.

Crossover:

1. Randomly determine a sub-tree of parent 1.
2. Copy the sub-tree to the offspring.
3. From parent 2:
   Visit remaining nodes in priority order. Insert them to offspring:
   - As deep as possible.
   - According to the BL-rule.
   - Take boundary constraints into account.
4. Assign priorities to the nodes from parent 2 in offspring: New values, smaller than those occurring so far.

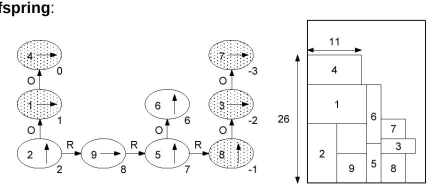
2-dimensional packing (4)

Crossover example: parents



2-dimensional packing (5)

Offspring:



2-dimensional packing (6)

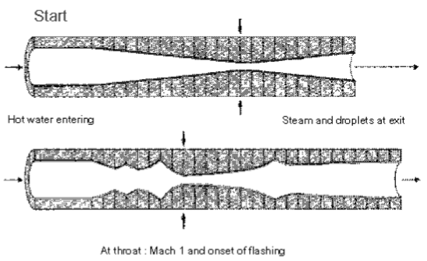
Objective function:

$$f = \frac{c}{h \cdot c + w} \to \max$$

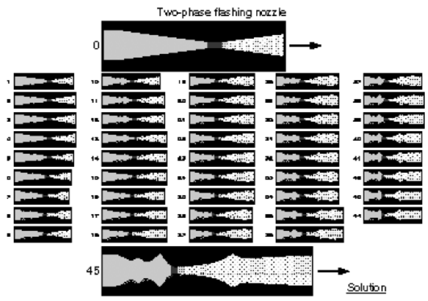
- c: a scaling integer value.
- For identical h values: prefer smaller w.

Remarks:
- A very challenging problem
- Traditional algorithms are not applicable for n > 30
- EA requires a *decoder*: trees $\to$ packing patterns
- EA solved problems up to n = 70

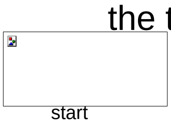
Hot water flashing nozzle (1)



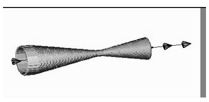
Hot water flashing nozzle (2)

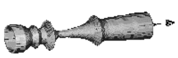


Demonstration 1:
the two-phase nozzle



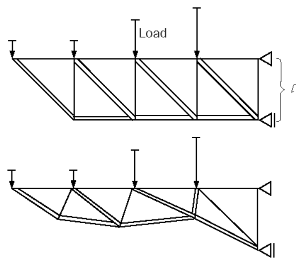
start





end

Minimal weight truss layout



Start

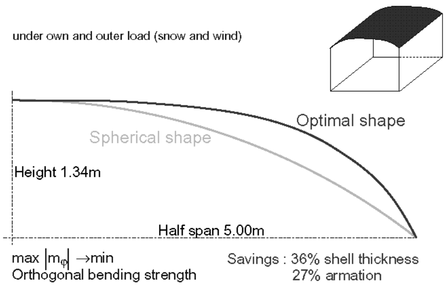
922 kp

(LP minimum)

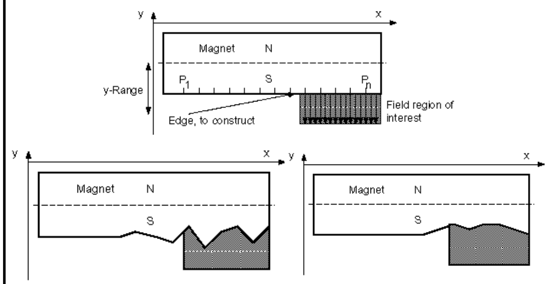
Optimum

738 kp

Concrete shell roof



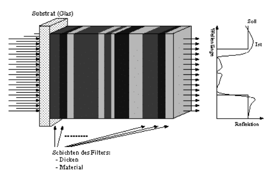
Dipole magnet structure (1)



Dipole magnet structure (2)

- Analysis of the magnetic field by Finite Element Analysis (FEM).
- Minimize sum of squared deviations from the ideal.
- Individuals: Vectors of positions (y_1, …, y_n).
- Optimization by an evolution strategy
- Upper graphic: standard solution
- Bottom left: 2.7% better than upper graphic
- Bottom right: 9.82% better than upper graphic

Optic Multilayer (1)



Problem parameters:
- Thicknesses $\bar{d} = (d_1, \ldots, d_n)$ of layers.
- Layer materials $\bar{\eta} = (\eta_1, \ldots, \eta_n)$ (integer values).
- Number of layers n.

⇒ Mixed-integer, variable-dimensional problem.

Optic Multilayer (2)

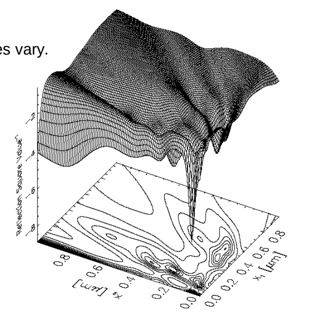
Objective function:

$$f(\bar{d},\bar{\eta}) = \int_{\lambda_d}^{\lambda^u} \left[R(\bar{d},\bar{\eta},\lambda) - \tilde{R}(\lambda) \right]^2 d\lambda$$

- $R(\bar{d},\bar{\eta},\lambda)$:
  Reflection of the actual filter for wavelength λ.
  Calculation according to matrix method.
- $\tilde{R}(\lambda)$: Desired reflection value.

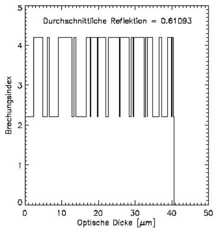
Optic Multilayer (3)

Example topology:
- n = 2.
- Only layer thicknesses vary.

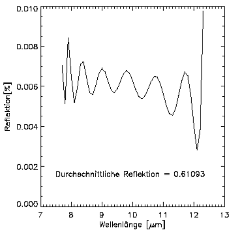


Optic Multilayer (4)

Example structure:



Example reflection:



Evolutionary Computing EA application examples 31

Optic Multilayer (5)

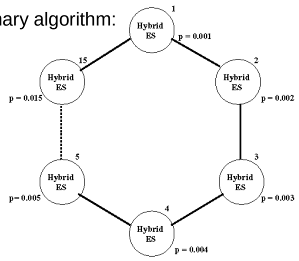
Existing Methods:

- Refinement methods:
  - Initial design constructed by an expert.
  - Local optimization of the initial design.
- Synthesis methods:
  - Without initial design (random start).
  - Automatical global optimization.

Evolutionary Computing EA application examples 32

Optic Multilayer (6)

Parallel evolutionary algorithm:

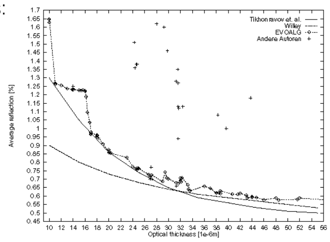


- Per node: EA for mixed-integer representation.
- *Isolation* and *migration* of best individuals.
- Mutation of discrete variables: Fixed p_m per population.

Evolutionary Computing EA application examples 33

Optic Multilayer (7)

Reference results:



Comparison of literature results, theoretical predictions, and EA results.

⇒ Excellent algorithm for the synthesis of filters.

Evolutionary Computing EA application examples 34

Remarks

- Complex practical problems
- Lack of (good) analytical/heuristic methods
- Hybridization with heuristics helps
- EAs can outperform well trained, experienced humans

Evolutionary Computing EA application examples 35