European Safety, Reliability and Data Association

46th ESReDA Seminar
May 29-30, 2014, Politecnico di Torino, Torino, Italy

IMPROVED METAHEURISTIC APPROACH TO SKEW REINFORCEMENT OPTIMIZATION IN CONCRETE SHELLS

Simona Mancini[a], Gabriele Bertagnoli[b], Luca Giordano[b]

[a] DIST, Control and Computer Engineering Department, Politecnico di Torino
[b] DISEG, Structural Engineering Department, Politecnico di Torino

Outlines

- Optimization and structural design
- Problem presentation
- Method description
- Real case analysis and computational results
- Conclusions and future developments

Structural design issues

Many decisions to be taken during the design phase

- Deck shape
- Reinforcement
- Pier shape

Many constraints to be satisfied

- Safety standard
- Physical constraints
- Loading constraints

Relation between stuctural design and optimization

- Designers make decisions in order to provide solutions which respect all the problem constraints

but, is the solution found a good solution?

could we do something better?

could we save money, materials, exc..?

could we reduce impact ?

We need to use optimization to answer all these questions!!

How to formulate an optimization problem ?

- Variables definition
- Solutions representation
- Solutions space definition (values which variables can assume)
- Objective function (goal)
- Solutions quality measure criteria
- Constraints expletation
- Distinction between compulsory and optional constraints

If constraints and objective function are linear and the number of variables is not too large

↓

The optimization model can be solved by means of commercial solvers

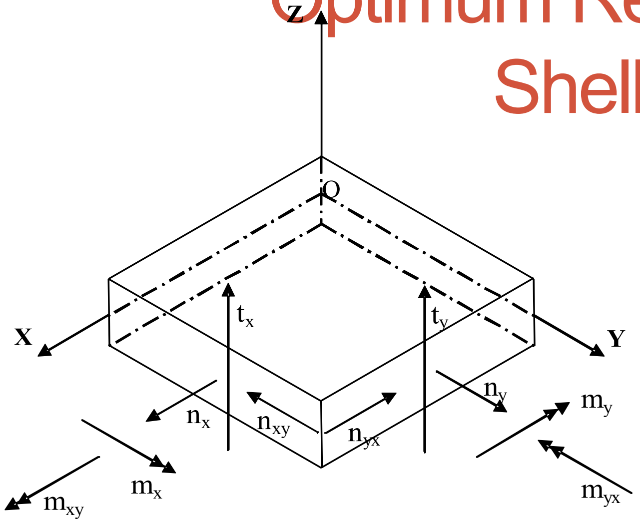
otherwise

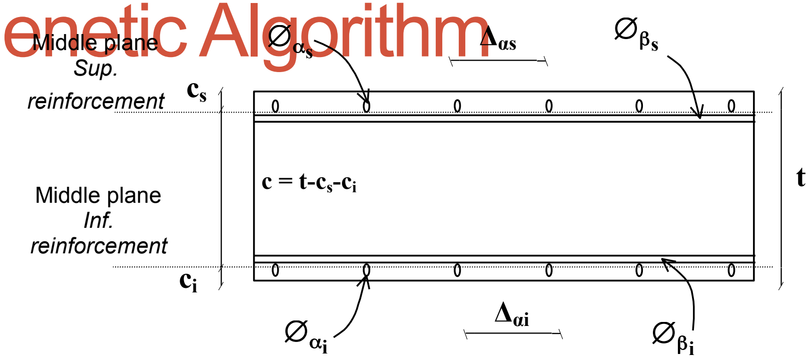
↓

Solvers are not suitable

Heuristics algorithms are needed

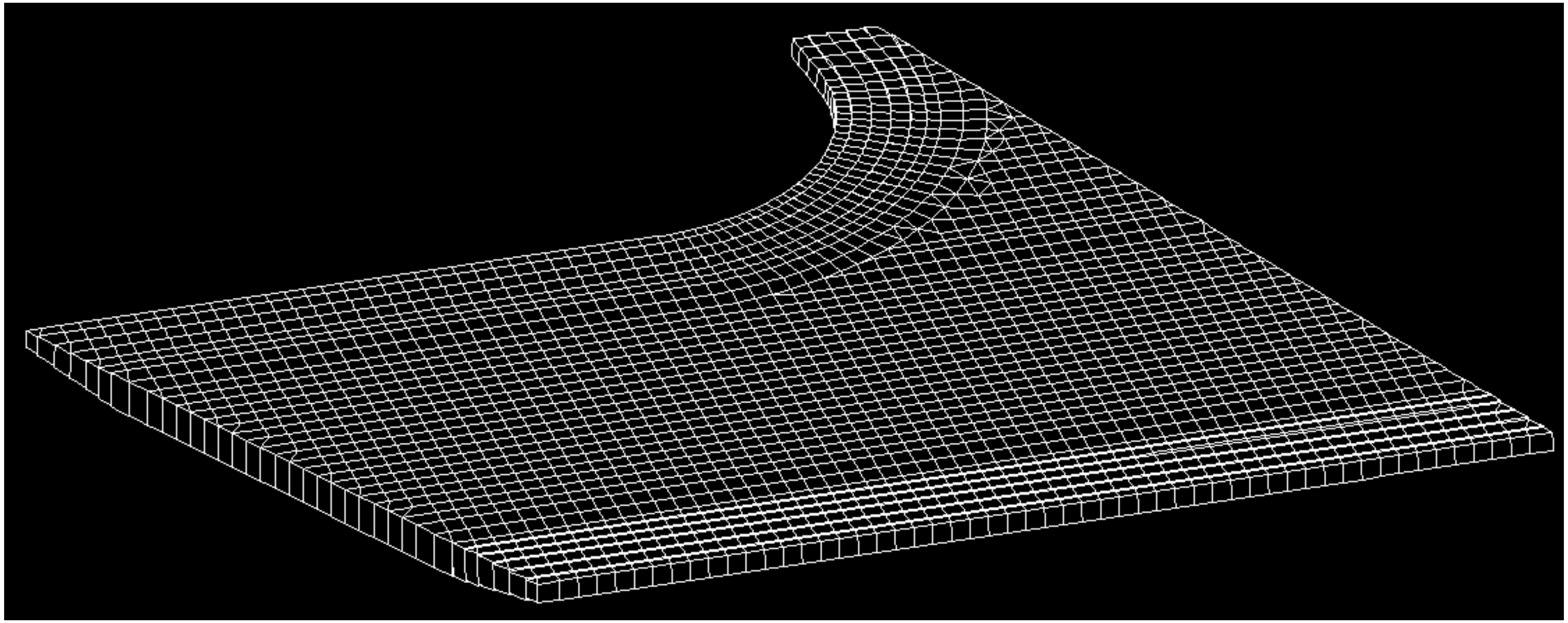
Optimum Reinforcement Design of Concrete Shell Elements by means of a Genetic Algorithm





Internal actions:
Nx Ny and Nxy from the plate
Mx My and Mxy from the slab
Tx and Ty plane shear components

3D mesh view (Shell Elements)



Structural behavior may be complex even if for simple structures

F.E.M Analysis

Design by means of finite elements (2D/3D) in cracked concrete and plastified renforcement

Solution representation and objective function

Solution can be represented as an array

$[\xi_s, \xi_i, \theta_s, \theta_i, \theta]$

For each combination of values we can obtain a global reinforcment value: A_{skr}

Solution quality can be measured as $A_{max} - A_{skr}$

Goal is to minimize A_{skr}

Problem Definition: variables

- layers thickness ξ_s and ξ_i
- angles θ_s , θ_i, inclination of concrete compression field in each layer
- angle θ inclination of concrete compression field in internal layer

All these variables can assume real values

There are physical constraints which must be respected

Problem Definition: constraints

- Maximum and minimum reinforcement range: $A_{min} < A_{skr} < A_{max}$
- Layer thickness domain:

$$2c_r \le \xi_r \le Ht \quad (r = s, i)$$

- Angle domain:

$$0 \le \theta_r \le \pi \quad (r = s, i)$$

- Global thickness amount:

$$\xi_s + \xi_i \le Ht$$

- Angle domain:

$$21.8 \le \theta \le 45$$

- Mechanical model constraints

Objective function: minimize the total reinforcement required to satisfy all the constraints

Problem solution

- The model is not linear and not convex due to the mechanical constraints

- Exact methods are not suitable

- Heuristics are required!

What are heuristic algorithms ?

- Algorithm to found feasible solutions and to improve it
- The goal is to find a good solution, not the optimal one, and even if we found it we cannot prove its optimality
- Fast and suitable even for large complex problems
- Can address non linear constraints and objective functions and multi-objective functions

A powerful tool for solving real problems

Optimization under uncertain loading conditions I

- What happens if we don’t know a priori the internal actions working on the shell element?
- We analyze the worst cases: octuples in which an action is minimum or maximum
- We should analyze the possible combinations of internal actions and use the minimum reinforcement required to satisfy all the combinations
- But how can we find it?

Optimization under uncertain loading conditions II

Keeping the highest value required by a combination in each direction

Not so smart!

Can we do something better?

LOAD CASE		$A_{s\alpha s}$ [cm²/m]	$A_{s\beta s}$ [cm²/m]	$A_{s\alpha i}$ [cm²/m]	$A_{s\beta i}$ [cm²/m]	$\frac{A_{sw} \cdot t}{s}$ [cm²/m]	A_{sTOT} [cm²/m]
1	N_{22} max	**70.57**	15.00	69.28	15.00	14.28	187.70
2	N_{33} max	15.00	37.06	15.00	18.55	0.01	85.63
3	N_{23} max	15.00	25.71	15.00	20.21	12.08	91.03
4	M_{22} max	49.51	15.00	76.45	15.00	13.16	172.41
5	M_{33} max	68.09	15.11	65.47	15.00	13.71	180.80
6	M_{23} max	15.00	35.29	15.00	15.97	0.01	81.26
7	V_{12} max	15.37	24.73	15.00	15.00	8.43	80.64
8	V_{13} max	16.18	29.21	15.00	**25.29**	13.70	102.81
9	N_{22} min	15.00	37.06	15.00	18.55	0.01	85.63
10	N_{33} min	70.57	15.00	69.28	15.00	**14.28**	187.70
11	N_{23} min	46.38	15.00	70.45	15.00	0.01	146.85
12	M_{22} min	15.00	25.61	15.00	18.69	0.01	74.31
13	M_{33} min	15.00	**37.10**	15.00	15.00	0.01	82.12
14	M_{23} min	70.46	15.00	69.23	15.00	14.10	187.30
15	V_{12} min	44.46	15.00	68.05	15.00	0.01	142.52
16	V_{13} min	52.26	15.00	**76.66**	15.00	14.22	176.70
SOLUTION		70.57	37.10	76.66	25.29	14.28	227.48

A new optimization problem

- New objective function

$$ATOT = \sum_{j=1}^{5} (max_i(A_{ij}))$$

$$ATOT = \sum_{j=1}^{5} A_{l^j j}$$

where l^j is the leader case for the j^{th} component of the reinforcement.

A heuristic framework I

- We focus our attention only on a direction at a time
- We analyze the leading combination
- We try to minimize the reinforcement required in the given direction avoiding the reinforcement required in the other directions, for the global problem, to grow up
- We try to avoid the leading combination to become leading for other directions

A heuristic framework II

Algorithm 1 **A heuristic framework (HEUR) for SRD2D under multiple loading conditions**

create an initial solution in which
$Z_j = max_i A_{ij} \quad \forall j \in [1, 5]$
find the leading case l_j such that
$l_j = i \rightarrow Z_j = A_{ij} \quad \forall j \in [1, 5]$
compute the objective function of the initial solution $A_{TOT}^{INIT} = \sum_{j=1}^{5} Z_j$
set the best solution objective function $A_{TOT}^{BEST} = A_{TOT}^{INIT}$
repeat
  for $1 \leq j \leq 5$ **do**
    run GA on case l_j minimizing $A_{ij} + \delta \sum_{k=1}^{5} (max(A_{ik}^{new} - Z_k; 0)$
    update the value of Z_k for each k
    calculate the new l_k for each k
    compute the value of the new objective function $A_{TOT}^{NEW} = \sum_{k=1}^{5} Z_k$
    if $A_{TOT}^{NEW} \leq A_{TOT}^{BEST}$ **then**
      $A_{TOT}^{NEW} = A_{TOT}^{BEST}$
      update the values of $A_{l^j k}$ for eacj k
    end if
  end for
until maximum number of generations, N_{ITER}, is reached or no improvements has been found since last iteration

Solving the subproblems

- If we analyze a given combination of internal actions
- We can solve it with an «ad hoc» heuristics
- We choose a genetic algorithm

Genetic Algorithms (GAs) quick overview

- Based on DNA structure
- Solutions encoding
- Initial population composed by N individuals
- Each individual has a reproduction probability based on its fitness (quality measure function)
- New individuals can be generated by the union of two or more individuals (may results to be very different from the parents- diversification) or by mutation of a single individual (it may results similar to the parent-intensification)
- Population can be completely regenerated at each iteration (enhanced diversification but possible loss of good information) or just a percentage of the individuals can be replaced (avoiding loss of good information but there is a risk of early convergence)

How GA works?

1. Initial Population creation

2. Introduction of a fitness function (representing solution quality) which indicate the probability of reproduction of an individual (solution)

3. Evolution process

Crossover operators (acting on two individuals)

Mutation operators (acting on a single individual)

4. New population

Restart from 2.

Genetic algorithm description: Initial Population

- Initial population should be heterogeneus but should contain good solutions
- Due to the nature of the problem there are some values of variables which yield to bad solutions whichever values take the other variables

1. We create 5 sets of N solutions each in which only one variable changes and the others are kept constant
2. We keep only good solutions (< 1.5*best)
3. We create initial population of size R combining one solution for each set

Reproduction probability

The probability for an individual *i* to be chosen for reproduction is given by:

$$f(X_i) = \frac{z_i}{\sum_{i=1}^{R} z_i}$$

being

$$z_i = \frac{1}{A_{SKRi}}$$

Population Evolution

- N_iter generations are created
- At each generation population is fully replaced
- At each step of the replacement an individual is randomly selected with the roulette wheel method
- If the chosen individual *i* has a $f(X_i)$ over the average (1/R)

good solution $\longrightarrow$ information to be preserved

we apply mutation

Otherwise we have a bad solution, we need to destroy and reconstruct the solution

  - We draw another individual and apply crossover

Mutation operator

- Apply a small mutation (perturbation) on each variable
- P = [p_1 p_2 p_3 p_4 p_5] generate Q = [q_1 q_2 q_3 q_4 q_5]

- q_i is randomly draw, in the interval [$a_{i,}$ b_i]

$$a_i = \max(c_i, p_i - \lambda_i) \qquad b_i = \min(p_i + \lambda_i, d_i)$$

Crossover operator

- $P = [p_1\ p_2\ p_3\ p_4\ p_5]$ and $Q = [q_1\ q_2\ q_3\ q_4\ q_5]$ generate a new individual

  $S= [s_1\ s_2\ s_3\ s_4\ s_5]$

- A crossover point *k* is randomly draw (variables values before *k* are taken from P the others from Q)
- Example k=3 S=[p1,p2,q3,q4,q5]

Crossover shakes elements from both parents

Attempt to combine good elements in order to create a "better" individual

Computational results

- Five cases
- Different values of delta

δ/instance	1	2	3	4	5
0	22.75	27.19	24.79	38.28	17.96
0.1	22.74	27.19	24.55	38.06	17.71
1	22.52	26.77	24.09	37.27	17.71
10	22.36	25.73	24.61	37.28	17.71
100	22.56	25.90	24.56	37.40	17.71
1000	22.39	25.69	24.55	37.43	17.71
Initial sol	22.75	27.19	24.79	38.28	17.96
improvement	**1.69%**	**5.50%**	**2.85%**	**2.64%**	**1.38%**

Conclusions I

- Optimization is a useful and necessary tool in structural design expecially in problem in which
  - There are different feasible solutions
  - We need to respect constraints but we also have to take into account solution quality
- Genetic algorithms fit well with this kind of problem
- Using optimization we can strongly reduce reinforcement with a huge save of materials and money and without loosing safety

Conclusions II

- If we are working under uncertain loading conditions (as it often happens) we can use the above presented heuristic framework to further improve the solution

Future developments

- Three dimensional elements
- Metaheuristics approach
  - Variable Neighborood Search
  - Memetic algorithms
  - …
- Closed Form for the single case optimization to be included as a balck box in the heuristic framework

Thank you for the kind attention