IEEE | Congress on Evolutionary Computation
5-8 JUNE, 2017 | SAN SEBASTIÁN (SPAIN)

Multiobjective Sizing Optimization of a **Steel Girder Bridge** with a Simple **Target-Driven Approach**

Authors

Mariapia Marchi¤, Luca Rizzian¤, Stefano Costanzo¤†
¤ ESTECO S.p.A., Area Science Park, loc. Padriciano 99, 34149 Trieste, Italy {marchi,rizzian,costanzo}@esteco.com www.esteco.com
† Department of Engineering and Architecture, University of Trieste, Trieste, Italy

Abstract

We present a simple strategy for multiobjective target-driven optimization and apply it to the sizing optimization of a steel girder bridge. Users or **decision makers** are asked to express their preferences (based on their previous experience) in terms of **desired target objective values** to **drive the optimization** towards the preferred regions of the Pareto front. This can lead to a **more efficient exploration** of specific regions of the objective space and **reduce the computational cost** of finding desirable solutions. This strategy combines **a-priori** approaches with **interactive preference-handling** approaches. These methods have recently received more attention in the evolutionary multiobjective optimization community. The proposed algorithm is described in detail and compared with existing methods. Benchmarks on standard mathematical test functions as well as on a realistic structural engineering sizing optimization problem are provided.

Algorithm

```
TARGET POINT
Interaction with the user:
for all objectives f_i do
  Provide target objective value t_i
  Provide tolerance percentage value Δf_i
end for
Provide desired number of points in the
hypervolume N_t
Initialize the counter of interesting
points to c = 0
- Phase 1:
At each evaluation the objective
function is modified
for all objectives f_i do
  o_i = g(f_i − t_i)
  given the function modulus g(x) = |x|
  if o_i < t_iΔf_i then
    Mark this solution as interesting
  end if
end for
if solution is interesting for each
objective then
  c = c + 1
end if
if c ≥ N_t then
  Switch to Phase 2
end if
- Phase 2:
Override g(x) = x
At each evaluation the objective
function is modified
for all objectives f_i do
  o_i = g(f_i − t_i)
end for
```

TARGET-POINT APPROACH

Aim: keep **high selection pressure in a preferred objective space area** without wasting computational time when too many objectives for thorough exploration.

Interaction with the user: user is asked to provide an a priori **target value** for each objective, **tolerance values** that define a hypervolume in the objective space and a target **number of points** N_t to be found inside the tolerance hypervolume

Phase 1 (reaching specified target): objective search space is modified while preserving its **multiobjective** nature and **original constraints**. Solutions ordered using Pareto sorting criterion on **modified objectives**. Phase 1 finishes when N_t is found

Phase 2 (attaining the true front): When the algorithm has met the target criteria, it starts a **classic evolution** by searching for the real Pareto front until the number of generations is completed

This approach is a kind of module that can be plugged into an existing algorithm. Here, it is applied to NSGA-II [1] and MOGA-II [2] to **prevent natural tendency of genetic algorithms to be driven by the simplest objectives** rather than those preferred by the decision makers (DM). DM could be more interested in a **specific region** of the search space instead of in a free evolution: this justifies **Phase 1.** If DM's knowledge of the problem search space is incomplete, an automatic optimization can find **unexpected solutions**: this justifies **Phase 2.** To summarize, the target-point module can work either **with Phase 1 alone** (reaching the target) or **with Phase 1 followed by Phase 2** (reaching the target and improving it further). Depending on the problem difficulty, Phase 2 can be **triggered** by appropriately setting tolerance and N_t (in addition to the GA number of generations).

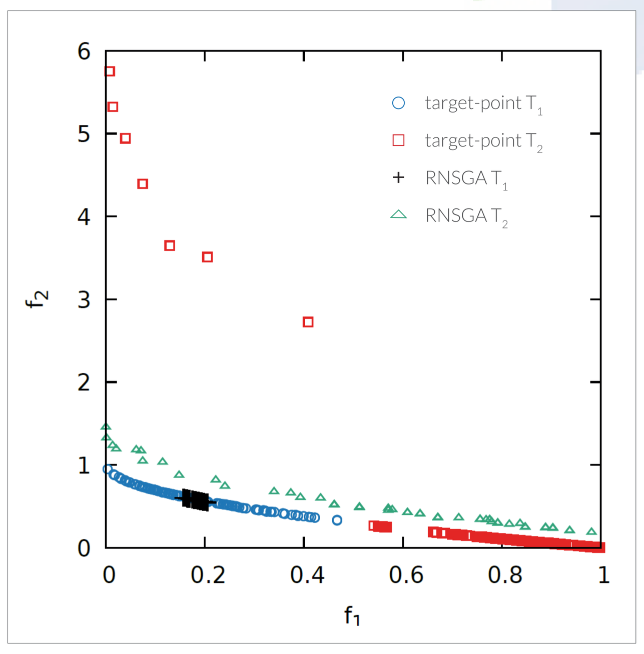
DATA ANALYSIS: visual comparison of solutions found with different algorithms. Known difficulty of a single performance index to account for all quality aspects of a set of solutions of a multiobjective optimization problem. Here even worse because of the contrasting nature of the two phases. Further investigations are required.

Test Problems

Problem	n	Range	Objective functions f_1, f_2	Constraint function c
ZDT1	30	$x_i \in [0,1]$, $i = 1, ..., n$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})[1 - \sqrt{x_1/g(\mathbf{x})}]$, $g(\mathbf{x}) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$	Unconstrained
CTP7*	4	$x_1 \in [0,1]$ $x_i \in [-5,5]$ $i = 2, ..., n$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})[1 - f_1(\mathbf{x})/g(\mathbf{x})]$, $g(\mathbf{x}) = 31 + \sum_{i=2}^{n}[x_i^2 - 10\cos(2\pi x_i)]$	$c(\mathbf{x}) = \cos(\theta)[f_2(\mathbf{x}) - e] - \sin(\theta) f_1(\mathbf{x}) \geq$ $a\lvert\sin\{b\pi[\sin(\theta)(f_2(\mathbf{x}) - e) + \cos(\theta) f_1(\mathbf{x})]^c\}\rvert^d$, $\theta = -0.05\pi$, $a = 40$, $b = 5$, $c = 1$, $d = 6$, $e = 0$

Table 1

ZDT1: TARGET-POINT vs. RNSGA [3] prototype [4]



For each algorithm, two optimizations with target values T_1 = **(0.2, 0.6)** and T_2 **(0.9, 0.1)** very close to true Pareto front. Both algorithms are run for **100 generations** starting from initial population of **50 random individuals.**

Target-point approach: tolerances = 2% and N_t = 50.

RNSGA prototype: extent control parameter = 0.001 and default weight vector values (equal weights).

T_1: both algorithms achieve desired target. Target-point approach also enters Phase 2, while RNSGA stays close around reference point.

T_2: RNSGA does not reach reference value within number of generations. Target-point approach reaches target and begins exploring further, although a number of spurious non–dominated solutions remain for small values of f_1 (some from initial population).

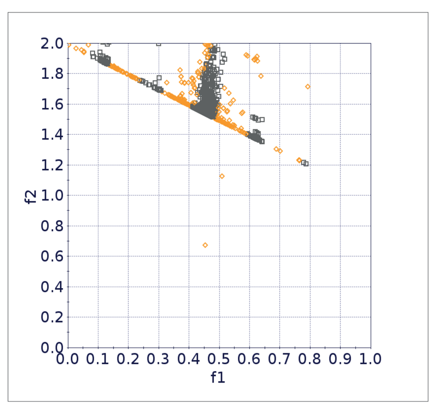
CTP7*: weighted sum vs. target point (0.5,0.5)

Comparison between **target-driven approach** (target-point MOGA-II) and linear aggregation of the objective values into a **weighted sum** (MOGA-II single objective).

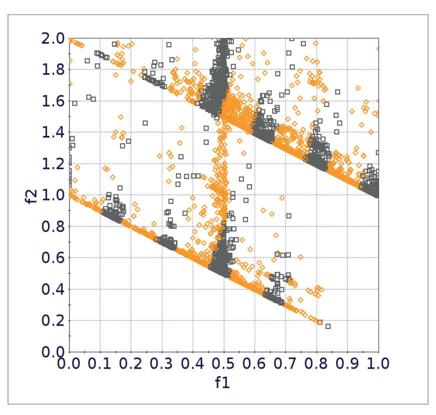
Both optimizations start from same initial population of **100 elements** created with uniform latin hypercube sampling and **250 generations**.

The **aim** is to get **equally important objective** values. **Weighted sum**: sum of the two objective values. **Target-driven optimization**: target values f_1 = f_2 = 0.5 (tolerance 1%) and N_t = 100.

Target-point effects cannot be reproduced by appropriately setting the weights in the weighted sum method. By aggregating objectives into a single-objective, objective search space is modified. Detrimental effect in the presence of constraints. Weighted sum follows a constraint in a sub-optimal region and gets stuck in local optimal solutions for the original optimization problem. Target-driven mechanism instead preserves multiobjective nature of original problem while trying to push optimization in the desired direction. Exploration is performed front-wise and eventually reaches the target.



Weighted sum



Target point

Application Case

MULTIOBJECTIVE SIZING OPTIMIZATION OF A STEEL GIRDER BRIDGE UNDER PERMANENT, VARIABLE AND SEISMIC LOADS: **discrete input variables, discretized objective space, several constraints to comply with specific performance and safety levels required by national and international laws and standards.**

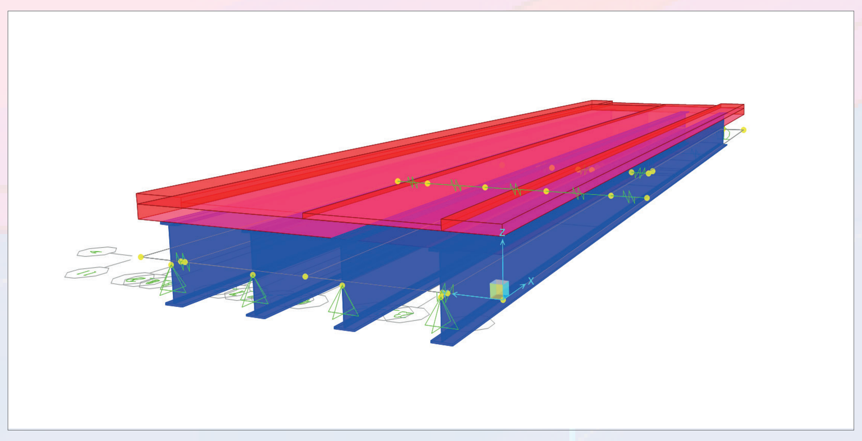
Bridge consists of four steel H-girders with flanges welded to the web. Girders support a reinforced concrete slab, an asphalt layer and two sidewalks. Bridge modeled with finite-element structural analysis software **SAP2000**. Resistance checks: vertical loads in ultimate limit state. Deformation checks: serviceability limit state. Seismic analysis: linear dynamic with response spectrum. Life-safety seismic project response spectra for the location site of Reggio Calabria (Italy).

Input variables: Table 2
Input constraints: Two constraints on web and flange values to exclude too slender sections of class 4
Output constraints:
- Flexural verification: girder bending moment smaller than resistance moment
- Shear verification: shear stress smaller than shear resistance
- Shear instability of the girder sections
- Serviceability Limit State compliance checks on vertical displacement of most deformed girder

Objectives (to minimize):
- Girders total weight (weight difference of 50 KN means 12000 € cost difference)
- Horizontal seismic shifts across the bridge (impact on maintenance cost)
- Vertical seismic shifts (impact on maintenance cost)

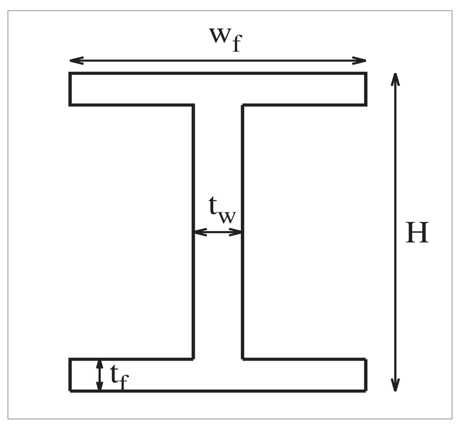
Input variables	Range (m)	Step (m)	# of values
Girder section height	[0.3 – 1.5]	0.01	121
Web thickness	[0.02 – 0.04]	0.001	21
Flange thickness	[0.02 – 0.04]	0.001	21
Flange width	[0.2 – 0.5]	0.005	61
Distance from edge	[0.35 – 1.5]	0.15	9

Table 2



SIMULATION AND RESULTS

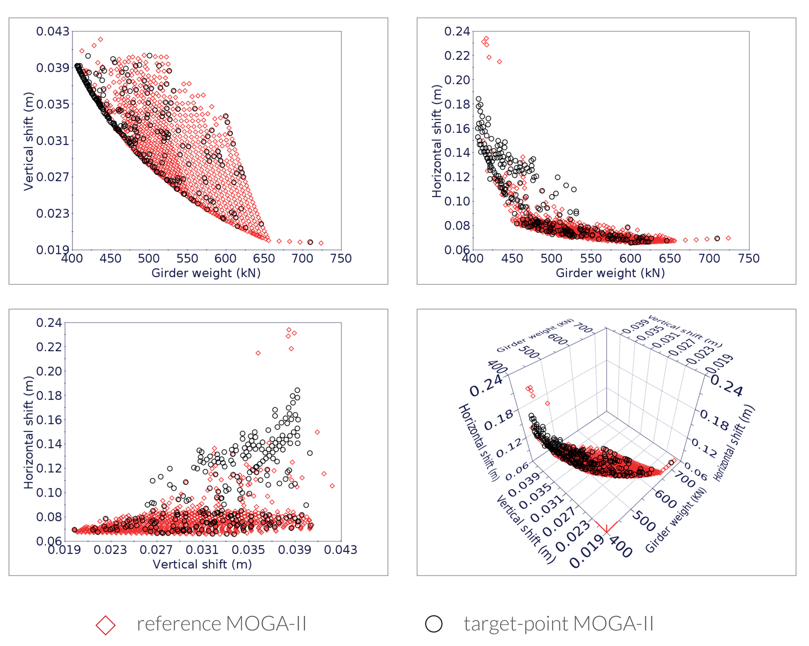
Reference optimization with MOGA-II:
Initial population of 100 random designs, **120 generations**, default optimization parameters
Result: compromise solutions between small seismic horizontal displacements and small weight values (i.e. between small initial and maintenance costs). Region of small horizontal shifts seems easier to explore. A **gap** can be noticed around horizontal shift = 0.18 m



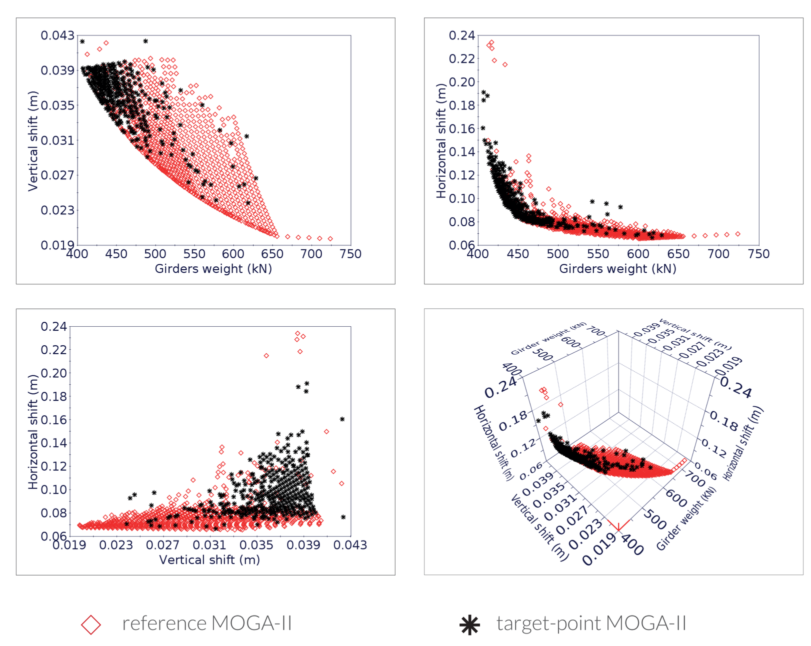
Target 1:
Goal: fill the gap in horizontal shift values found with reference optimization
Target parameters: horizontal shift = 0.18 m (tolerance 1%), weight = 402 KN (tolerance 1%), vertical shift = 0.038 m (tolerance 5%), N_t = 100
60 generations, initial population of 100 designs selected from reference simulation optimal solutions (94 randomly chosen + 6 selected close to target region)
Result: several optimal solutions found in target objective-space region

Target 2:
Goal: find compromise solutions quickly
Target parameters: horizontal shift = 0.08 m (tolerance 1%), weight = 402 KN (tolerance 1%), vertical shift = 0.038 m (tolerance 5%), N_t = 100
60 generations, same initial population as for reference optimization
Result: target-point approach successfully reaches target region and begins exploring neighboring regions (Phase 2)

Bridge Target 1: reference MOGA-II vs. target-point MOGA-II



Bridge Target 2: reference MOGA-II vs. target-point MOGA-II



References

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197, 2002
C. Poloni and V. Pediroda, "GA coupled with computationally expensive simulations: tools to improve efficiency," in Genetic Algorithms and Evolution Strategies in Engineering and Computer Science, D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, Eds. New York: John Wiley & Sons, 1997, ch. 13, pp. 267–288
K. Deb, J. Sundar, Udaya Bhaskara Rao N., and S. Chaudhuri, "Reference point based multi-objective optimization using evolutionary algorithms," International Journal of Computational Intelligence Research, vol. 2, no. 3, pp. 273–286, 2006
S. Lin. NGPM, a NSGA-II program in Matlab v1.4. [Online]. Available: https://it.mathworks.com/matlabcentral/fileexchange/31166-a-nsga-ii-program-in-matlab-v1-4