MDO RESEARCH GROUP
Multidisciplinary Design Optimization

Multi-disciplinary Optimization; Recent Developments and Challenges Ahead

Tapabrata Ray

MDO Group, UNSW, Canberra.

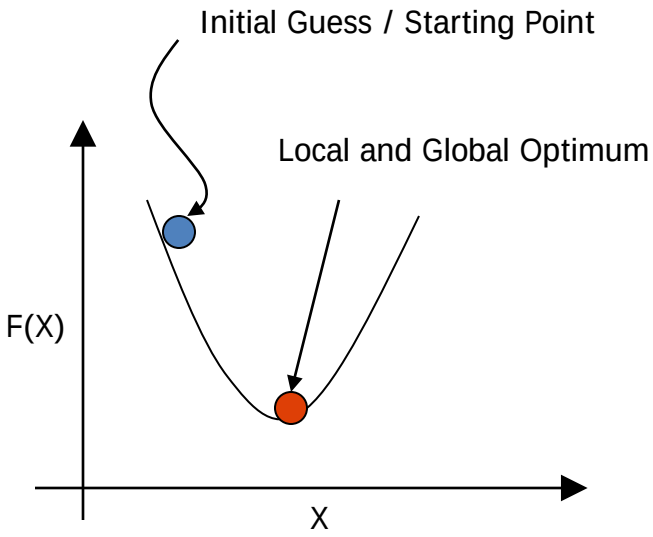
http://seit.unsw.adfa.edu.au/research/sites/mdo/index.htm

Acknowledgements: All former and current postgraduate students, collaborators and funding agencies.

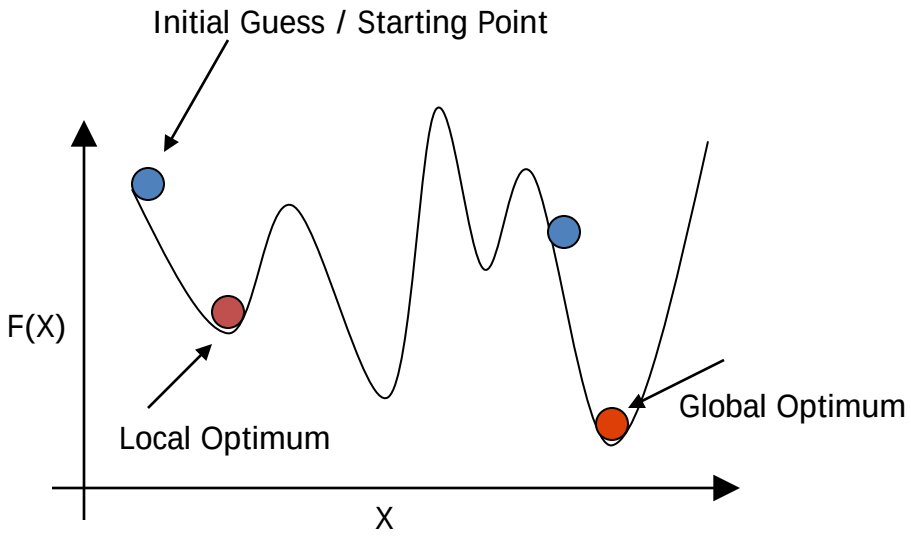
UNSW THE UNIVERSITY OF NEW SOUTH WALES

Overview and Basic Concepts

Basic Premise



Unimodal Function



Multimodal Function

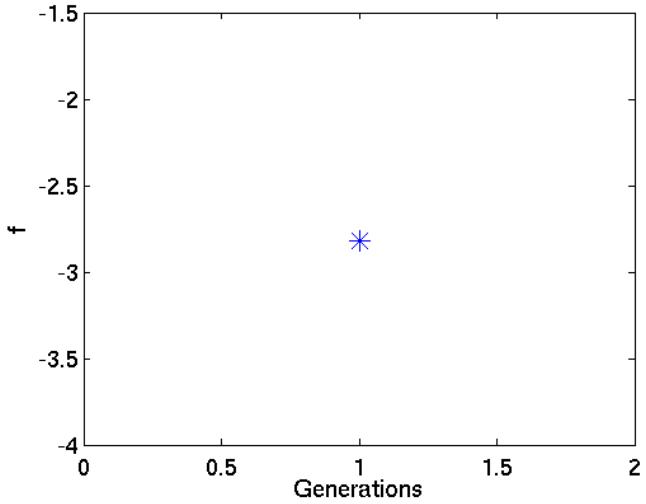
No Algorithm Can Guarantee to Locate Global Optimum for Multimodal Functions.

Gradient Based Algorithms Can Guarantee to Locate Local Optimum.

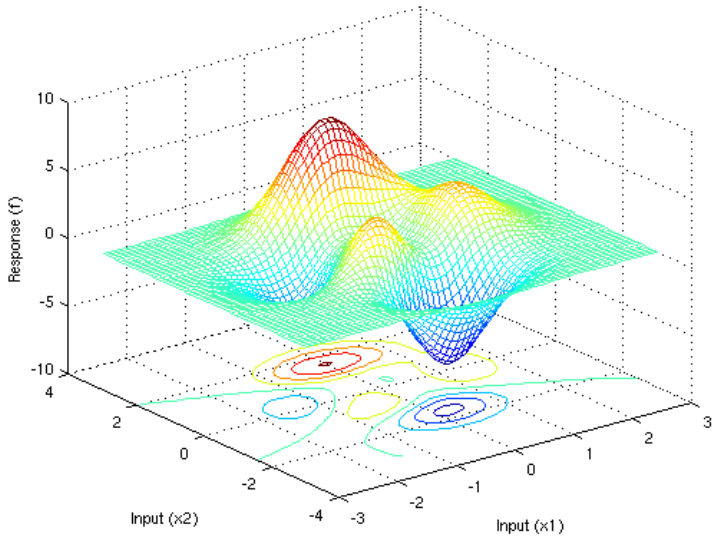
Zero Order Methods Can only Locate a *Good Solution* which may not even be a Local Optimum.

Stochastic Population Based Algorithms

1. Generate a set of M solutions.
2. Identify better solutions as parents.
3. Combine the parents to create M child solutions.
4. Combine the original set of solutions (M) and child solutions (M).
5. Select M solutions from the above set of 2M solutions.
6. Repeat steps 2-4 till convergence condition is true.

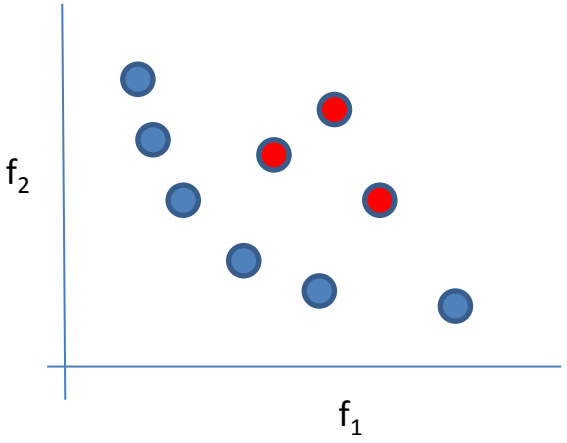


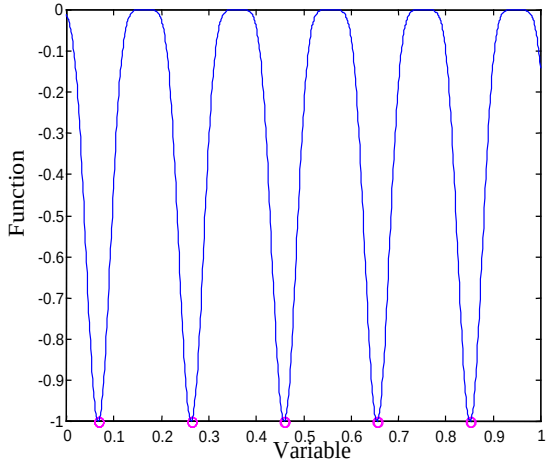
Progress Plot



Visualization

Multiobjective Optimization





Minimization and Maximization problems are interchangeable.

All discussions are in the context of minimization.

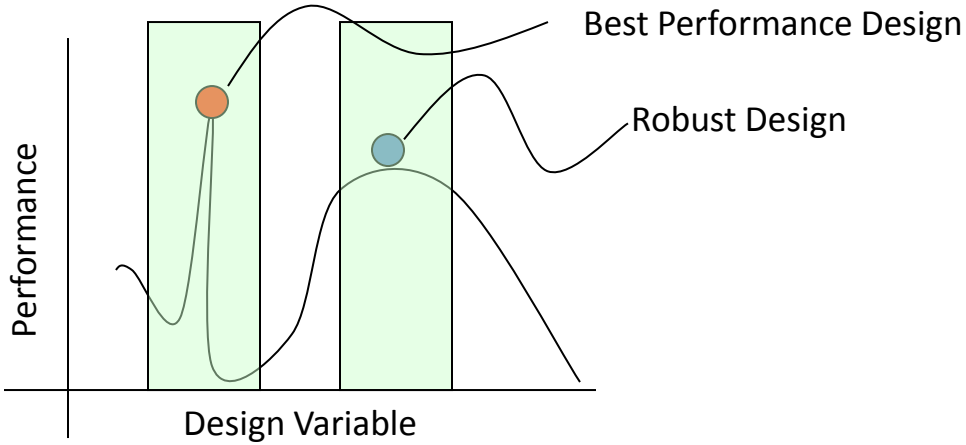
In Multio-bjective optimization, the interest is to find the non-dominated set of solutions that are close to the Pareto optimal set.

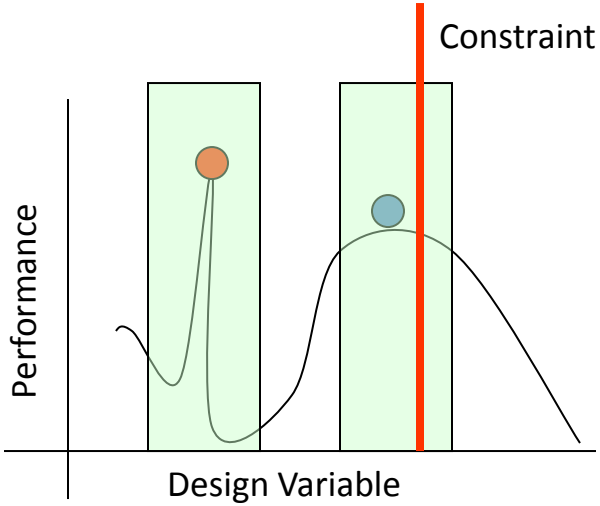
The ND set of solutions should have a good convergence and a good spread.

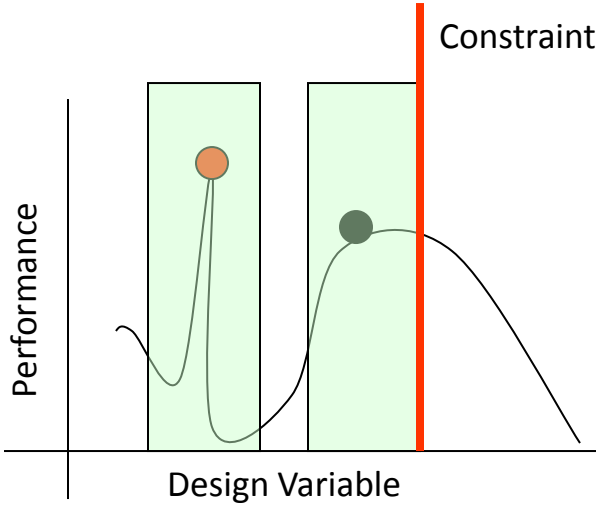
The solutions in blue are Non-dominated solutions. The solutions in red are dominated solutions.

Designers are interested in identifying multiple optima.

Basic Premise





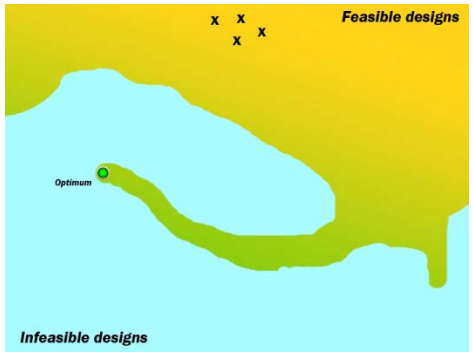


Notice that the Design Shifted away from the Constraint Boundary

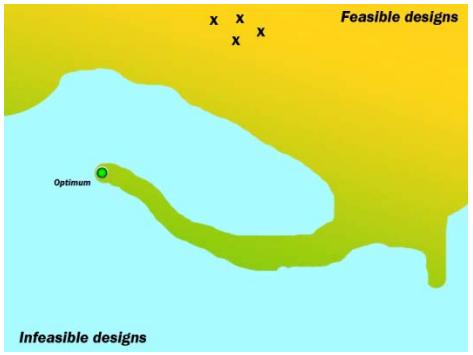
Constrained Optimization

Constraints and their Effects

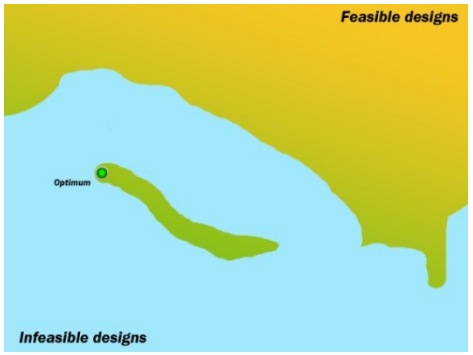
- Solutions to constrained optimization problems often lie on constraint boundaries.
- Most real coded population based stochastic algorithms intrinsically prefer a feasible solution over another.
- Proposed Infeasibility Driven Evolutionary Algorithm IDEA).



Real coded EA



Preserving Infeasible Solutions



Disconnected Feasible Regions

- Explicitly preserves a fraction of infeasible solutions across generations.
- Marginally infeasible solutions are preferred over feasible solutions.
- Offers a trade-off set of solutions with minimal constraint violation in addition to the set of feasible solutions.

Infeasibility Driven Evolutionary Algorithm (IDEA)

Infeasibility preservation
- Parameter α determines the ratio of infeasible solutions in the population

Optimization problem reformulation
- Constrained, k objectives → Unconstrained, k+1 objectives
- Additional objective = constraint violation measure

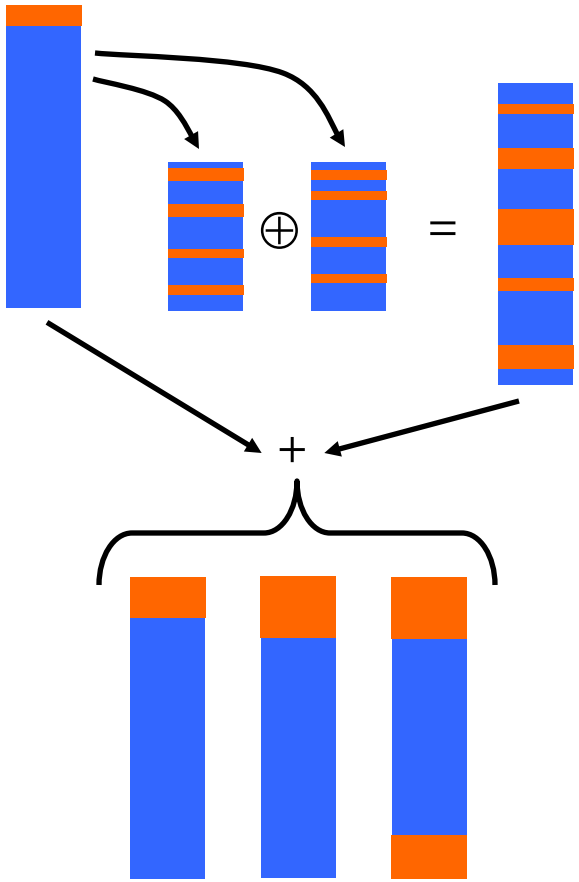
Rank "good" infeasible solutions higher than feasible solutions
- Active search through feasible as well as infeasible design space

- All solutions in the population are assigned a rank corresponding to each constraint.
- Solutions that do not violate a particular constraint are assigned rank 0 for that constraint.
- Rest of the solutions are assigned ranks in increasing order of constraint violation.
- Violation measure for a solution is calculated as the sum of individual ranks corresponding to all constraints.

Soln.	Violation Value			Relative Ranks			Violation measure
	C1	C2	C3	C1	C2	C3	
1	3.50	90.60	8.09	3	7	7	17
2	5.76	7.80	6.70	4	5	5	14
3	-	3.40	7.10	0	3	6	9
4	1.25	-	0.69	1	0	1	2
5	13.75	90.10	5.87	6	6	4	16
6	100.70	2.34	3.20	7	2	2	11
7	-	5.09	4.76	0	4	3	7
8	1.90	-	-	2	0	0	2
9	-	110.56	-	0	8	0	8
10	8.90	2.30	9.80	5	1	8	14

Infeasibility Driven Evolutionary Algorithm (IDEA)

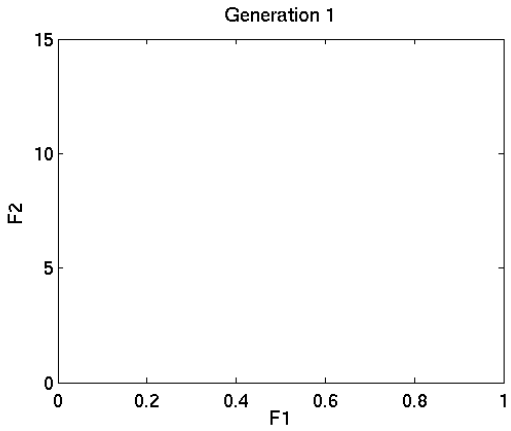
- Binary tournament based Parent selection
- Crossover and Mutation to create offspring population
- Non-dominated sorting (feasible and infeasible)
- Elite preservation (retain infeasible)



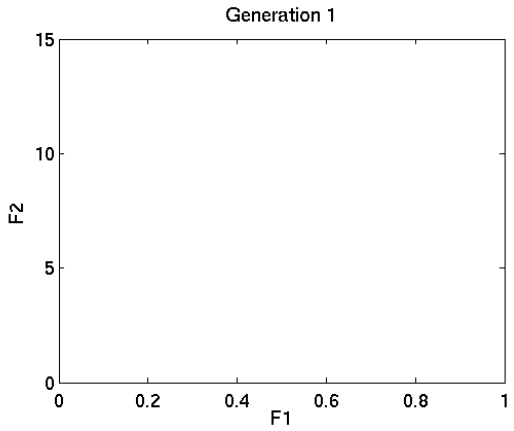
- Take note the infeasible solutions are on the top.
- Can only work if there are infeasible solutions, else its identical to a real coded elitist GA.
- Three possibilities of a population state is presented at the bottom.

Performance of IDEA

- Behaviour of IDEA on multi-objective optimization problems.



Real coded EA



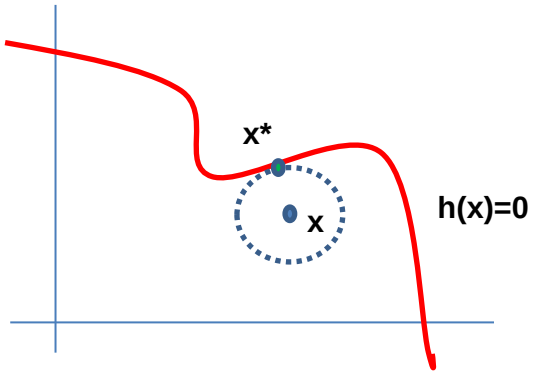
IDEA

Ray, T., Singh, H.K., Isaacs, A., and Smith, W. Infeasibility Driven Evolutionary Algorithm for Constrained Optimization, *Constraint-Handling in Evolutionary Optimization*, *Studies in Computational Intelligence Series 198*, Eds, Efrén Mezura-Montes, Springer. pp 145-165., 2009.

Singh, H.K. , Ray, T. and Sarker, R. ,"Optimum oil production planning using infeasibility driven evolutionary algorithm," Evolutionary Computation, In Press, (Accepted 09/201 1).

Equality and Active Inequality Constraints: Most Probable Point

The proposed method belongs to the class of repair, wherein infeasible solutions are repaired using the idea of *Most Probable Point (MPP)* (of failure). The notion of MPP is derived from *Reliability Based Optimization* (RBO).



x* is the MPP of x. For every solution x, there is a MPP corresponding to every constraint.

$$X_u = \frac{(X - \bar{X})}{\sigma_X}$$

X_u is in the transformed space.

$$||U|| = ||X^*{}_u - X_u||$$

For any infeasible solution, find MPP's for each violated constraint. This is solved using SQP.

Compute the overall constraint violation at each of the MPP's and pick the best and update x by x*.

The overall constraint violation could be the value of the maximum violated constraint or the sum of constraint violation.

Saha, A. and Ray, T. A repair mechanism for active inequality constraint handling, in Proceedings of the IEEE Congress on Evolutionary Computation, (Brisbane, Australia), pp. 1240-1247, 2012.

Saha, A. and Ray, T. Equality constrained multi-objective optimization, in Proceedings of the IEEE Congress on Evolutionary Computation, (Brisbane, Australia), pp. 47-53, 2012.

Handling Equalities using Most Probable Point

TABLE I

ATTAINING FEASIBLE STATE. (*#Succ* IS THE NUMBER OF SUCCESSFUL RUNS, I.E. IN WHICH THE FEASIBLE STATE WAS REACHED, WHEN THE SEARCH WAS TERMINATED, *#GenID* IS THE GENERATION IN WHICH THE FEASIBLE STATE WAS REACHED, *#Func. eval.* IS THE TOTAL NUMBER OF FUNCTION EVALUATIONS, *Hyper-area* IS THE HYPER-AREA AT THE FEASIBLE STATE.)

Prob.	**Algo.**	**#Succ.**	**#GenID**	**#Func. eval.** (Mean, Median)	**Hyper-area** (Mean)
czdt1	NSGA-II	30	293	1.1723e+04, 1.1700e+04	0.3517
	NSGA-II_MPP	30	**7**	**5.6849e+03, 5.8520e+03**	**0.6090**
czdt2	NSGA-II	30	293	1.1721e+04, 1.1720e+04	0.1259
	NSGA-II_MPP	30	**6**	**5.7188e+03, 5.8345e+03**	**0.2804**
czdt3	NSGA-II	30	292	1.1711e+04, 1.1700e+04	0.4626
	NSGA-II_MPP	30	**6**	**5.6077e+03, 5.4365e+03**	**0.8501**
czdt4	NSGA-II	30	418	1.6731e+04, 1.5520e+04	0.3317
	NSGA-II_MPP	30	**18**	**1.6007e+04, 1.6096e+04**	**0.4657**
czdt6	NSGA-II	30	87	3.4973e+03, 3.4800e+03	0.0369
	NSGA-II_MPP	30	**4**	**1.2789e+03, 1.2830e+03**	**0.2622**

TABLE II

INEQUALITY CONSTRAINED PROBLEMS WITH ACTIVE INEQUALITY CONSTRAINTS USING $N_{max} = 3X10^5$

Prob.	Algo.	#Succ.	Obj. Fn. (Best, Worst, Median)
g2	RGA	30	(**-8.0082e-01**, **-7.1466e-01**, **-7.8612e-01**)
	RGA_MPP	30	(-7.8708e-01, -6.6824e-01, -7.4896e-01)
g6	RGA	26	(-6.9291e+03, -6.4801e+03, -6.8866e+03)
	RGA_MPP	**30**	(**-6.9618e+03**, **-6.9618e+03**, **-6.9618e+03**
g10	RGA	30	(7.1747e+03, 1.1532e+04, 8.2717e+03)
	RGA_MPP	30	(**7.1538e+03**, **8.5009e+03**, **7.7754e+03**)
g24	RGA	30	(-5.5075e+00, -5.4713e+00, -5.4922e+00)
	RGA_MPP	30	(**-5.5080e+00**, **-5.5074e+00**,**-5.5080e+00**)

While existing methods try to minimize constraint violation, this approach attempts to minimize the distance. While the at the end, the solution has a CV equal to 0 and a distance =0, both traverse different landscapes.

Introduction

- The performance of all population based stochastic optimization algorithms are known to be affected by the presence of constraints

- The nonlinearity, multi-modality and the feasibility space associated with each constraint is likely to be different.

  Constraint handling methods can be broadly categorized in four different types:
  - use of penalty functions
  - repair schemes
  - use of decoders and
  - the separation of objective function and constraints

- However, in all such formulations *a full evaluation policy* is adopted wherein for an infeasible solution

- An important question is "why do we spend computational resources to evaluate constraints of a solution, when it has already violated a constraint ?".

Assuming that one is only interested in a feasible solution (preferably optimum) at the end of the search process, it is important to investigate the worth of evaluating infeasible solutions i.e. the cost of evaluation versus the knowledge gained to steer the search

Proposed Method

Step-1: Divide the population into (p+q) subpopulations , where p is the number of inequality constraints and q is the number of equality constraints.

Step-2: Assign a constraint sequence to each subpopulation i.e. (g_1 , g_2,……,g_p) for p number of inequality constraints

Step-3: The solution is migrated to a feasible subpopulation, when it satisfies all the constraints in the prescribed sequence.

Sequence sorting

	Constraint Sequence			Constraint Sequence			Constraint Sequence
	Initial order			Final order			
	g1	g2	g3	g1	g2	g3	
S3	0	0	1	0	0	1	S3
S2	0	3	_	0	3	_	S2
S4	2	_	_	2	_	_	S4
S1	5	_	_	5	_	_	S1

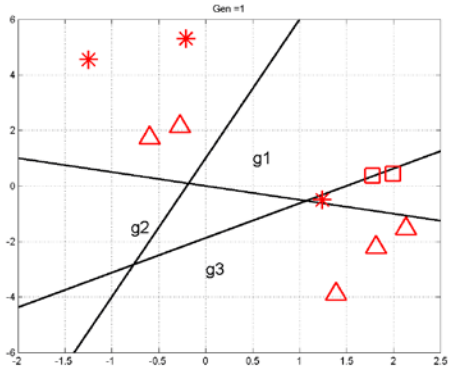
N.B. The constraint which has not been evaluated marked as '_'

- **The problem has three linear constraints and the optimum solution lies on a constraint boundary.**

$$\text{Minimize } f_1(x) = x_1^2 + x_2^2 + 2 * x_1 * x_2$$

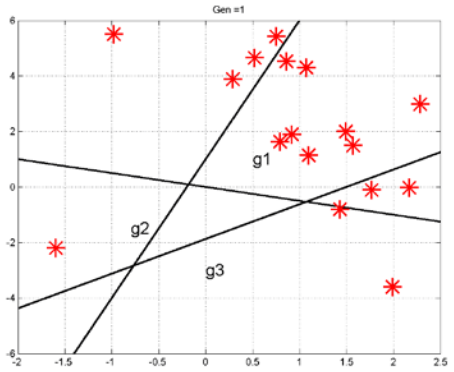
Subject to

$$g_1(x) \equiv x_1 + 2 * x_2 \geq 0,$$
$$g_2(x) \equiv 10 * x_1 - 8 * x_2 - 15 \geq 0,$$
$$g_3(x) \equiv -10 * x_1 + 2 * x_2 - 2 \geq 0,$$



Feasibility using Sequencing

- Continue evaluation of a solution as long as it satisfies the constraints.
- An efficient method of constraint sequencing to capture the feasible region from different directions



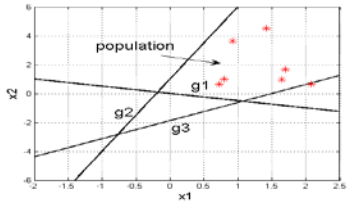
Feasibility using Violation

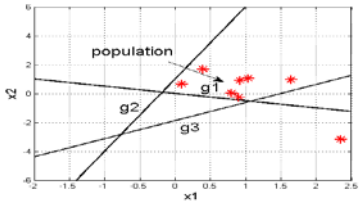
Asafuddoula, M. , Ray, T. and Sarker, R., "A self-adaptive differential evolution algorithm with constraint sequencing," in Lecture Notes in Computer Science, (Accepted 08/2012).

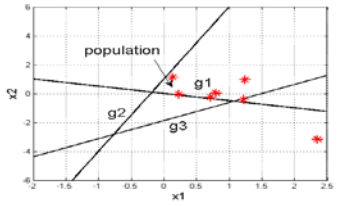
Example - 1

Trajectory of infeasible solutions

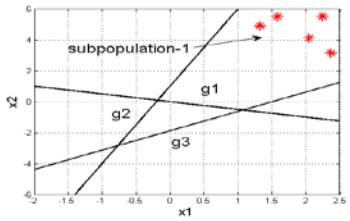
Progress of **population** Generations- 5 , 10, 20

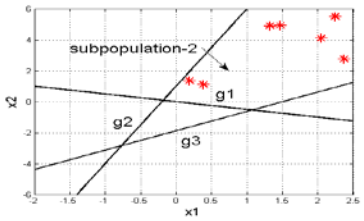


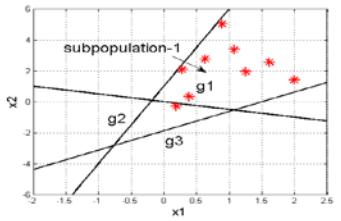




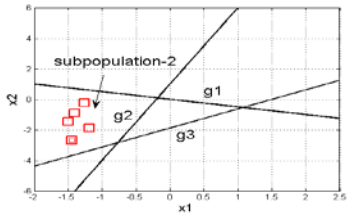
Progress of **subpopulation-1** Generations- 5 , 10, 20

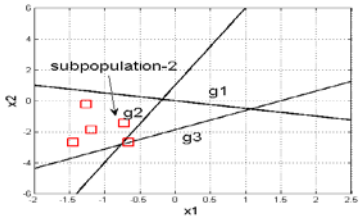


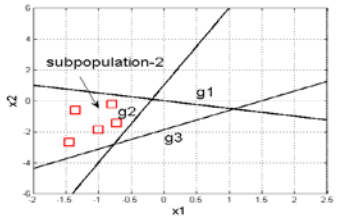




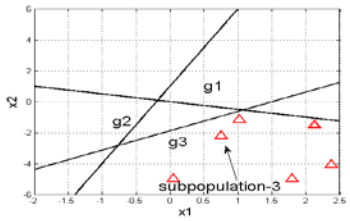
Progress of **subpopulation-2** Generations- 5 , 10, 20

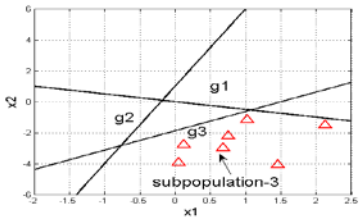


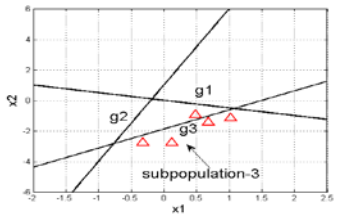




Progress of **subpopulation-3** Generations- 5 , 10, 20





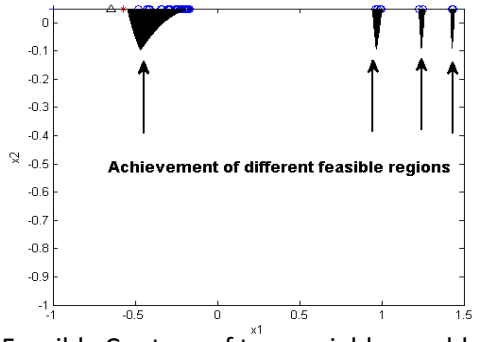


Example-2

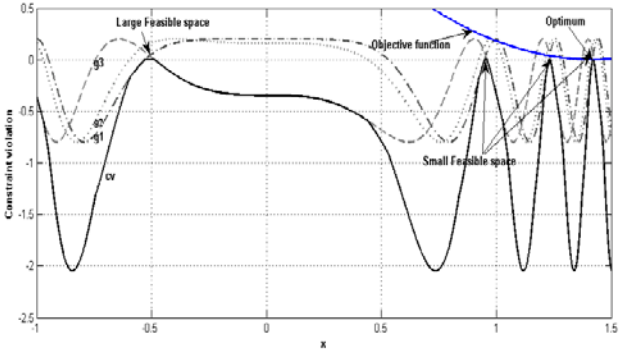
Problem with multiple disconnected feasible regions

- A two variables optimization problem involving three constraints is presented below
- The problem has four disjointed feasible regions of different sizes

Feasibility using Sequencing to achieve multiple disjoint feasible regions.

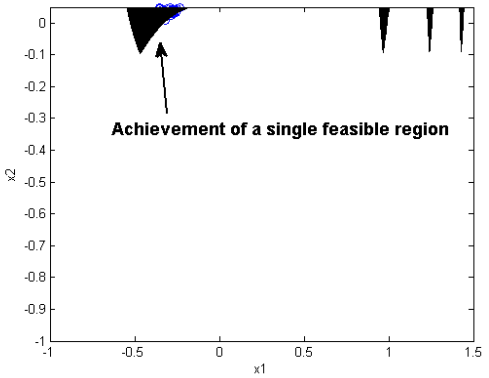


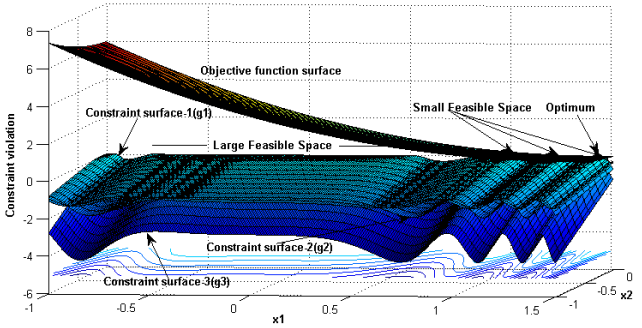
Feasible Contour of two variables problem



Single Variable problem

Feasibility using Violation which often trapped to a local optima.





Two Variables problem

Asafuddoula, A. , Ray, T. and Sarker, R., "A Differential Evolution Algorithm with constraint sequencing ," in Proceedings 3rd Global Congress on Intelligent Systems, (Wuhan, China) 2012

Comparison with Other Methods

Algorithms in comparison

- ε-DE
- JDE
- COPSO
- SaDE

In a comparison with ε-DE, the proposed algorithm is better in 7 out of 11.

Problem	D	Search Range	f type	No. of Inequality	Feasibility (ρ)
g01	13	$[0,1]^9[0,100]^3[0,1]^{(D-12)}$	quadratic	6	0.0111%
g02	20	$[0,10]^D$	nonlinear	2	99.971%
g04	5	$[78,102][33,45][27,45]^{(D-2)}$	quadratic	6	52.1230%
g06	2	$[13,100][0,100]$	cubic	2	0.0066%
g07	10	$[-10,10]^D$	quadratic	8	0.0003%
g08	2	$[0,10]^D$	nonlinear	2	0.8560%
g09	7	$[-10,10]^D$	polynomial	4	0.5121%
g10	8	–	linear	6	0.0010%
g12	3	$[0,10]^D$	quadratic	1	4.7713%
g18	9	$[-10,10]^8[0,20]^{(D-8)}$	quadratic	13	0.000%
g24	2	$[0,3][0,4]$	linear	2	79.6556%

Proposed Method, DEACS	Function eval.					
Prob.	Best		Median		Worst	
	#func.	#const.	#func.	#const.	#func.	#const.
g01	**6,847**	**10,537**	**11,699**	**16,787**	**16,613**	22,163
g02	**49,176**	**71,438**	105,373	152,686	**108,302**	154,151
g04	**3,691**	**4,857**	**4,234**	**5,592**	**4,777**	**6,110**
g06	**2,426**	6,057	**2,626**	7,820	4,162	9,214
g07	**12,652**	**26,790**	**17,891**	**37,084**	**47,587**	87,334
g08	138	900	557	1,537	812	1,960
g09	**6,608**	**9,291**	**7,418**	**10,547**	**8,109**	**11,931**
g10	**13,849**	**32,364**	**20,954**	**46,230**	**38,609**	**77,123**
g12	**450**	**450**	**1,000**	**1,000**	**1,200**	**1,200**
g18	**5,674**	**13,206**	**7,554**	**18,240**	**34,839**	**57,641**
g24	1,584	2,454	1,867	2,948	2,143	3,559
ε−DE	Function eval.					
Prob.	Best		Median		Worst	
	#func.	#const.	#func.	#const.	#func.	#const.
g01	18,594	18,594	19,502	19,502	19,971	19,971
g02	108,303	108,303	114,347	114,347	129,255	129,255
g04	12,771	12,771	13,719	13,719	14,466	14,466
g06	5,037	**5,037**	5,733	**5,733**	**6,243**	**6,243**
g07	60,873	60,873	67,946	67,946	75,569	75,569
g08	**621**	**621**	**881**	**881**	**1,173**	**1,173**
g09	19,234	19,234	21,080	21,080	21,987	21,987
g10	87,848	87,848	92,807	92,807	107,794	107,794
g12	2,901	2,901	4,269	4,269	5,620	5,620
g18	46,856	46,856	57,910	57,910	60,108	60,108
g24	**1,959**	**1,959**	**2,451**	**2,451**	**2,739**	**2,739**

Conclusion and Limitation

- The scheme has been embedded within a DE algorithm
- The schema offers great ability to achieve the feasible region from multiple feasible directions
- The scheme is generic and can be embedded within other forms of population based stochastic optimization algorithms
- Such an approach could be beneficial in two ways i.e. (a) less likely to be trapped at a local optima and

  (b) save computational cost by avoiding evaluation of infeasible solutions whenever infeasibility is detected

Limitations

- The constraints are evaluated in a cyclic order (i.e. sequence) in which the potential sequence could be missing and could be a limited usage.
- The number of sub-populations and the size of each sub-population are depended upon the number of constraints which could be non-trivial for handling large number of constraints.
- The proposed schema does not provide the best possible sequence out of the given constraint sequences.

Further Improvement

The proposed method is modified with the following steps:

Step-1: Assign random constraint sequences to every individuals in the population .

Step-2: Compute the number of satisfied constraints (NS) and the amount of violation(V).

Step-3: With the number of constraints satisfied taking a precedence over the violation value, a sorting is yield to the ranking of the individuals.

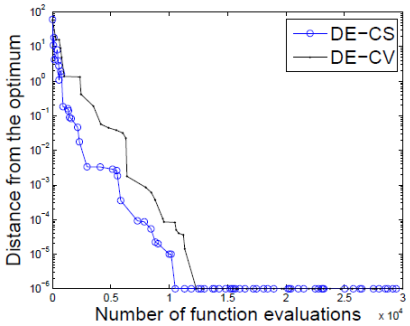
- For example assume a population, containing 4 solutions (S1, S2, S3, S4). The constraint violation matrix would assume a form illustrated in Table 1 with S3 identified as the best and S1 the worst

S1,S2,S3 and S4 are assigned with random constraint sequences to evaluate

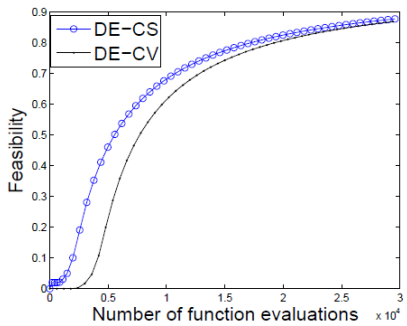
Initial order				NS	V	Final order
$S1\ (g_1, g_2, g_3)$	5	—	—	0	5	S3
$S2\ (g_2, g_3, g_1)$	0	3	—	1	3	S2
$S3\ (g_1, g_3, g_2)$	0	0	1	2	1	S4
$S4\ (g_2, g_1, g_3)$	2	—	—	0	2	S1

Example-3

- The feasible space of the problem is the feasible space dictated by constraint g3



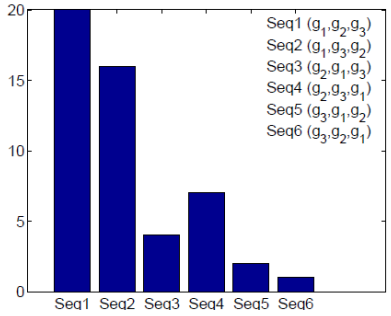
(a)

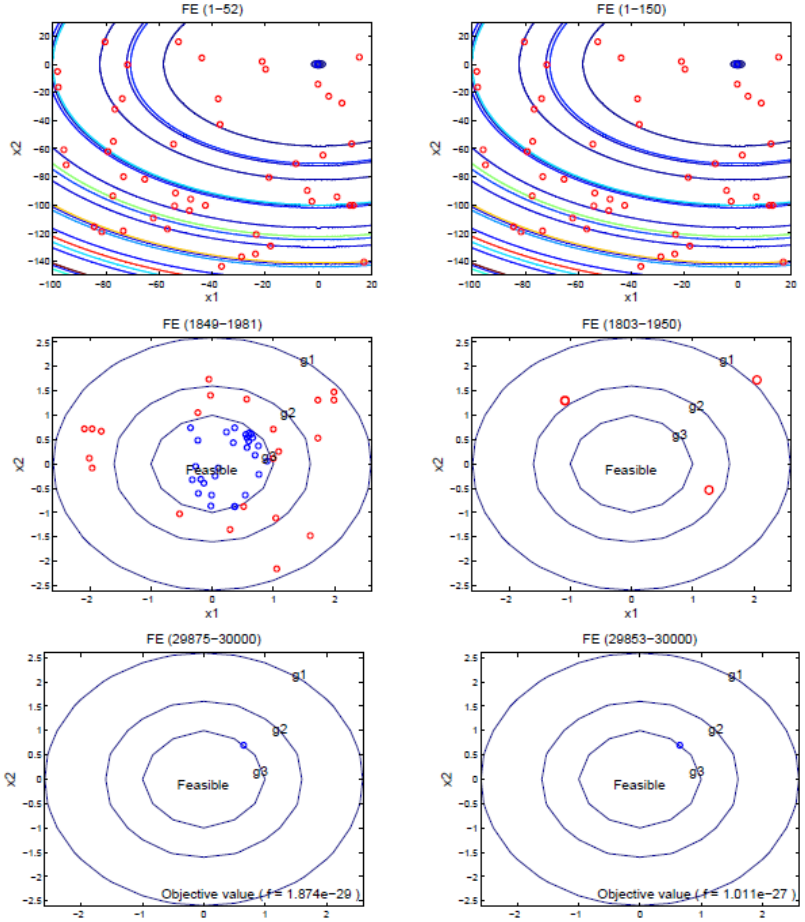


(b)

Progress plots (a) distance of a best feasible solution from the optimum (b) feasibility of the solutions

The most used constraint sequence is (g_1,g_2,g_3) and less used sequence is (g_3,g_2,g_1) out of 50 independent runs.



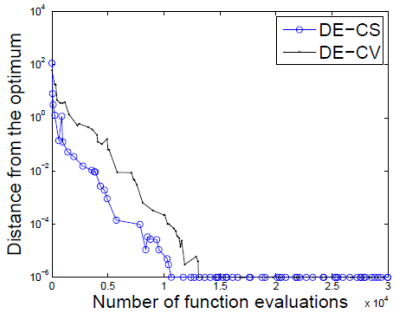


Progress of CS

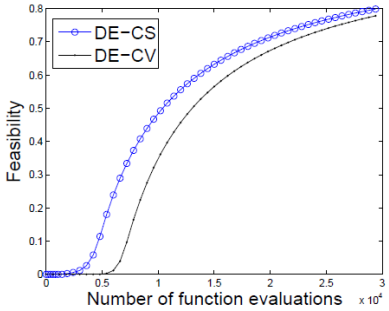
Progress of CV

Example-4

- The feasible space for the problem lies at the intersection of the feasible spaces of the individual constraints i.e. (i.e.g1∩g2∩g3)

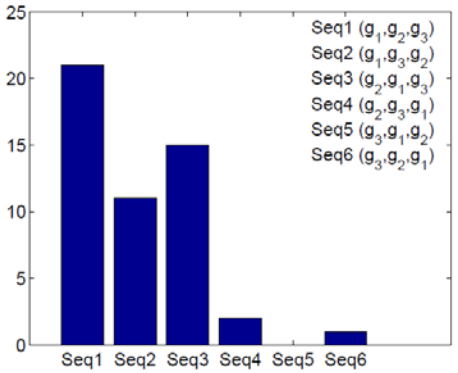


(a)

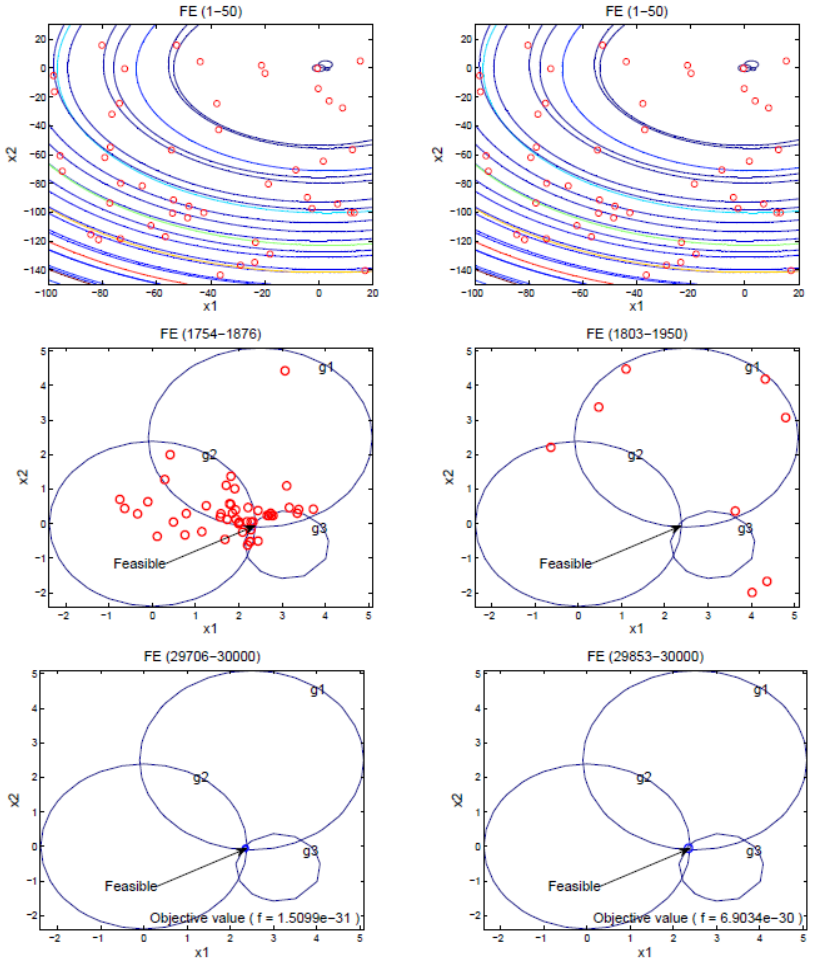


(b)

Progress plots (a) distance of a best feasible solution from the optimum (b) feasibility of the solutions



The most used constraint sequence is (g_1,g_2,g_3) and less used sequence is (g_3,g_1,g_2) out of 50 independent runs.



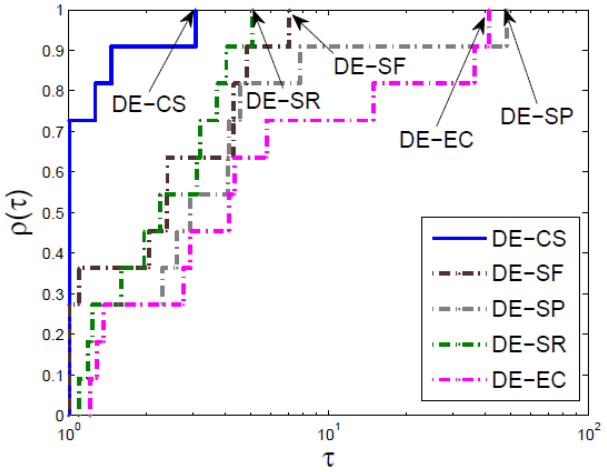
Progress of CS

Progress of CV

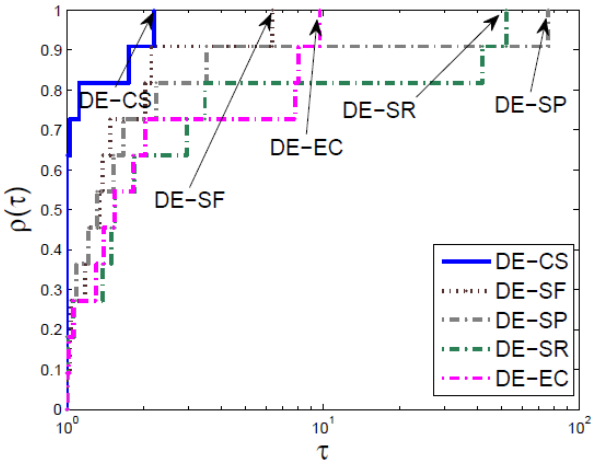
Comparison with Other Constraint Handling Methods

- We reported the results obtained by using stochastic ranking (SR) , self adaptive penalty (SP) , superiority of feasibility (SF) and epsilon constraint (EC) within the same framework of DE using *g-series* benchmark.

- The average number of function evaluations (*NFEs) required to* identify the first feasible solution and the computational time required to obtain the known optimum is plotted as a performance profile.



Performance profiles of DE-CS and others based on NFEs



Performance profiles based on computational time

- The results clearly indicate the superiority of DE-CS over other strategies in terms of obtaining the first feasible solution

Discussions

- The utility of using multiple constraint sequences is highlighted using two illustrative examples
- The results on the test problems clearly indicate that the approach is computationally efficient and better than existing strategies for constraint handling
- The proposed scheme delivers first feasible solution using less number of function evaluations in most of the test problems
- While identification of a feasible solution fast does not guarantee the search to deliver optimum solutions early, such a scheme is useful if one is interested in identifying a *feasible solution to a problem with limited computational* resource

Managing Multiple Objectives: Many Objective Optimization

Many objective optimization: Pareto Corner Search

- For more than four objectives, non-dominated sorting is ineffective.
- With a finite population, providing well spread and converged solutions on a hyper-surface is difficult.
- Some objectives might be redundant.
- Our Approach: Focus on Corner Solutions: Pareto Corner Search Evolutionary Algorithm

The key solutions proposed in this work are so called “Pareto corner solutions”

Consider an optimization problem with M Objectives, where the objective set is denoted by F_M. Now consider a subset F_k of the original objectives set, with k < M objectives. If minimization of k-objectives in this subset results in a single solution in the M objective space, then this solution is referred to as the corner(or Pareto corner) solution.

Proposed technique

- Instead of whole Pareto front, search for *key solutions* on it.
- These key solutions must exhibit good convergence and diversity.
- Analyze this set of solutions for identifying redundant objectives.

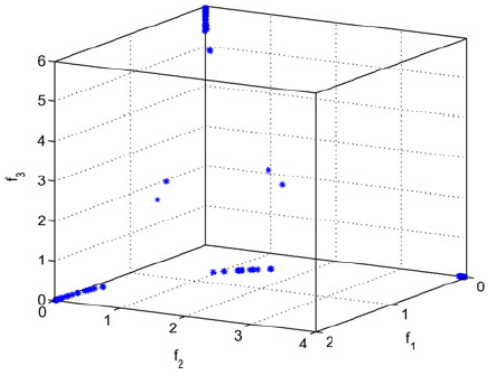


Fig. 9. Final population obtained for $WFG3_{conv1}$ using PCSEA.

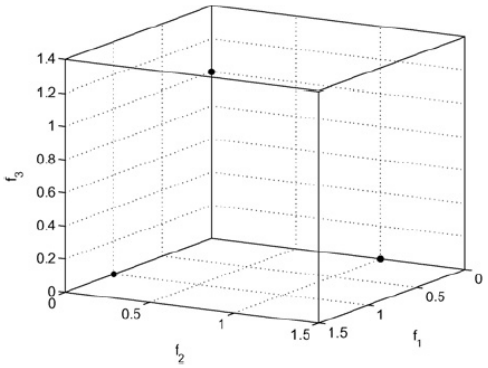


Fig. 6. Final population obtained for DTLZ2 using PCSEA.

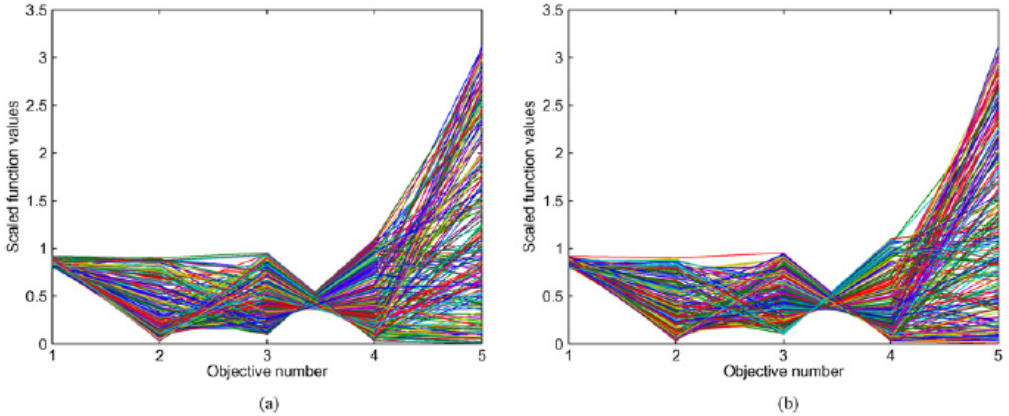
Performance of Pareto Corner Search

- Able to predict dimensionality accurately for up to DTLZ-(I,M) problems up to 100 Objectives.
- Up to DTLZ-(5,30) problems accurately Identified using less than 5 % of the function evaluations as used in the previous best reported studies.

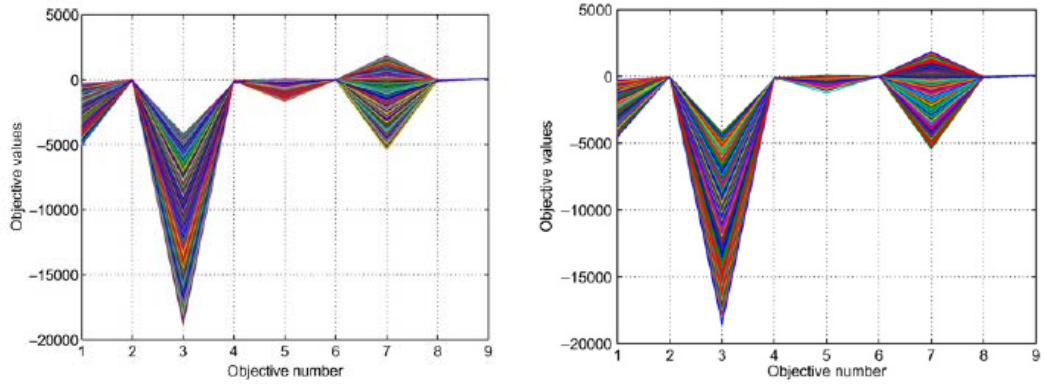
Savings in Computational Cost

Test Problem	Number of Evaluations	
	Earlier Studies	Present Studies
DTLZ5-(5-10) [12]	2 × 100 × 5000	200 × 200
DTLZ5-(5-20) [12]	2 × 100 × 5000	200 × 200
DTLZ5-(5-30) [12]	2 × 100 × 5000	200 × 200

Water resource problem (5 objectives, reduced to 3)



Radar waveform problem (9 objectives, reduced to 6)



Singh, H.K., Isaacs, A., Ray, T. (2011), A Pareto Corner Search Evolutionary Algorithm and Dimensionality Reduction in Many-Objective Optimization Problems, IEEE Transactions on Evolutionary Computation, Vol. 15, Issue 4, pp. 539 - 556.

Introduction

- Many objective optimization typically refers to problems with the number of objectives greater than four.
- The commonly used dominance based methods for multi-objective optimization, such as NSGA-II, SPEA2 etc. are known to be inefficient for many-objective optimization as non-dominance does not provide adequate selection pressure to drive the population towards convergence.
- There are also radically different approaches to deal with many objective optimization, such as attempts to identify the reduced set of objectives or corners of the Pareto front.

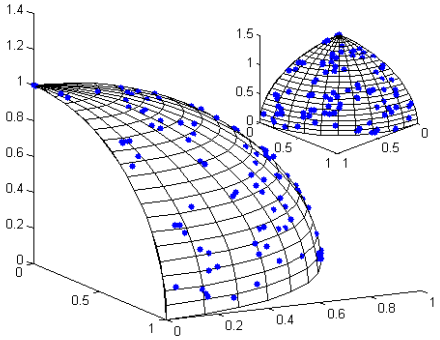
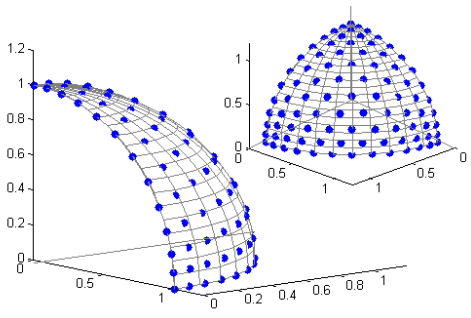


Fig: 1. Using Traditional Approach (NSGA-II)



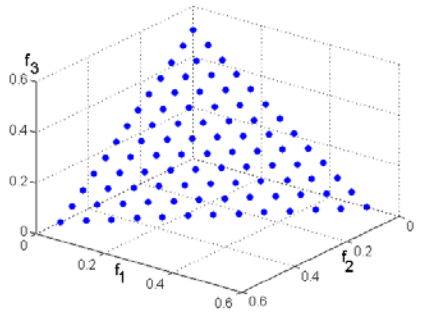


Fig: 2. Using Systematic Sampling

- Interactive use of decision makers preferences .
- Use of reference points from systematic sampling or solution of the problem as a hypervolume maximization problem.

Introduction

- Decomposition based evolutionary algorithms are yet another class of algorithms originally introduced as MOEA/D
- The multi/many-objective optimization problem is decomposed into a series of scalar optimization problems using different scalarization approaches (i.e. Weighted Sum Approach, Tchebycheff Approach or Normal Boundary Intersection Method)
- In the context of many objective optimization, the first issue relates to the design of a computationally efficient scheme to generate *W* uniform reference directions for a *M objective optimization problem*
- The second issue related to scalarization has been addressed via two fundamental means i.e. through a systematic association and niche preservation mechanism

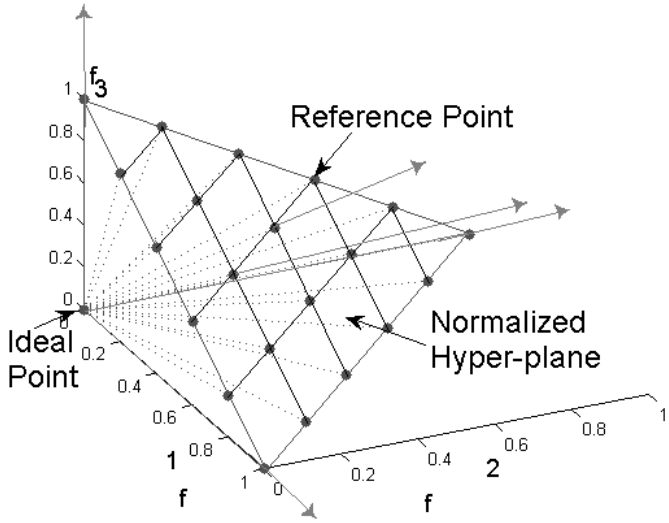


Fig. 3. A set of reference points in a normalized hyper-plane for number of objectives, *M = 3.*

Asafuddoula, A. , **Ray, T.** , Sarker, R. and Alam, K. "An adaptive constraint handling approach embedded MOEA/D," in Proceedings of the IEEE Congress on Evolutionary Computation, (Brisbane, Australia), pp. 2516-2513, 2012.

Proposed Method

The algorithm consists of four major components –

- generation of reference directions and assignment of neighbourhood
- computation of distances along and perpendicular to each reference direction
- method of recombination using information from neighbouring sub-problems and finally
- adaptive epsilon comparison to manage the balance between convergence and diversity.

Algorithm 1. DBEA-Eps

Input: Gen_{max} maximum number of generations, W the number of reference points

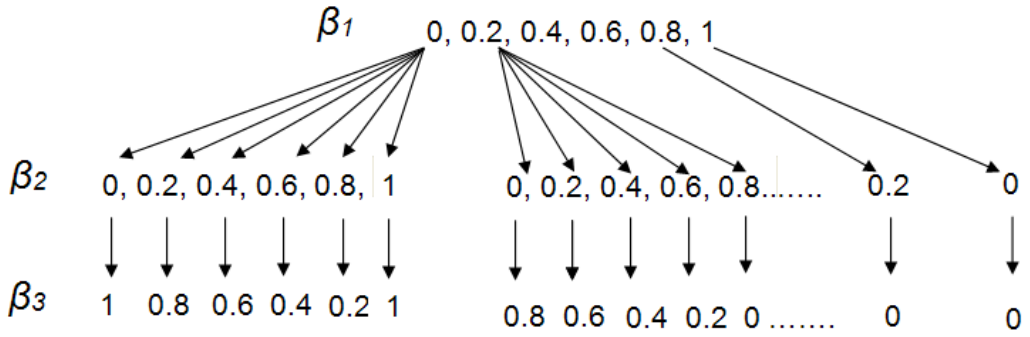
Generate the reference points and assign their neighborhood
Initialize the population P; $|P| = W$
Evaluate the initial population and compute the ideal point $\bar{z}_j = (f_1^{min}, f_2^{min}, ..., f_M^{min})$ and intercepts a_i's for $i = 1$ to M
Scale the individuals of the population
while $(gen \leq Gen_{max})$ **do**
    for *i=1:W* **do**
        Assign the base parent as P_i
        I=Select a mating partner for (P_i)
        Create a child via recombination as C_i
        Evaluate C_i and compute the distances ($d1$ and $d2$) using all reference directions
        Replace the parent P_k with C_i using *single-first encounter*, where k denotes the index of the first parent satisfying the condition of replacement
        Update the ideal point ($\bar{z}$), the intercepts and re-scale the population
    end for
end while

Proposed Method

➢ **Generation of reference directions and assignment of neighbourhood**

- A structured set of reference points (β) is generated spanning a hyper-plane with unit intercepts in each objective axis

- The approach generates *W points on the hyper-plane with a uniform spacing of δ = 1/p for any number* of objectives *M*



(a)

β_1	β_2	$(1-\beta_1-\beta_2)$
0	0	1.0000
0	0.2000	0.8000
0	0.4000	0.6000
0	0.6000	0.4000
0	0.8000	0.2000
0	1.0000	0
0.2000	0	0.8000
0.2000	0.2000	0.6000
0.2000	0.4000	0.4000
0.2000	0.6000	0.2000
0.2000	0.8000	0
0.4000	0	0.6000
0.4000	0.2000	0.4000
0.4000	0.4000	0.2000
0.4000	0.6000	0
0.6000	0	0.4000
0.6000	0.2000	0.2000
0.6000	0.4000	0
0.8000	0	0.2000
0.8000	0.2000	0
1.0000	0	0

(b)

Fig. 4. (a) the reference points are generated computing β's recursively (b) the table shows the combination of all β's in each column

The process of generation of the reference points is illustrated for a 3-objective optimization problem i.e. (*M=3) and with an assumed spacing of δ = 0.2* i.e. (*p = 5) in the Figure*

➢ **Computation of Distances along and Perpendicular to Each Reference Direction**

- The intercepts of the hyper-plane along the objective axes are denoted by $a_1, a_2, ..., a_M$. *The* generic equation of a plane through these points can be represented using the following equation

$$Af_1 + Bf_2 + ... + Cf_M \qquad \text{----------(1)}$$

where, *A, B,....,C are the unit normal of the plane. The intercepts of the plane with the* axis are given by $a_1 = 1/A$, $a_2 = 1/B$,....., *and* $a_M = 1/C$.

- An example of intercepts computation for a two-objective problem

A and **B** are the unit normal along the axes

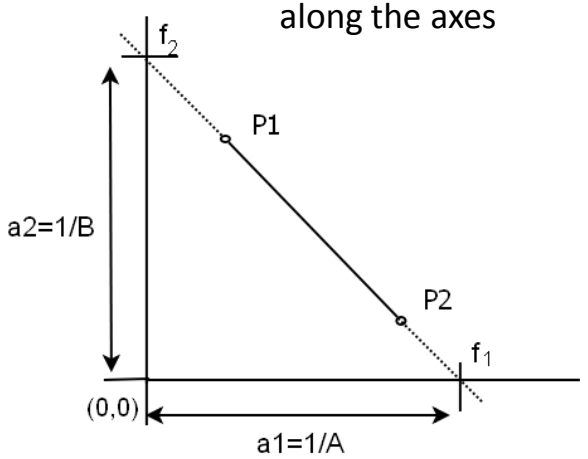


Fig. 5. Computation of intercepts of a two-objective problem

Every solution in the population is subsequently scaled as follows:

$$f_j'(x) = \frac{f_j(x) - z_j}{a_j - z_j}, \forall j = 1, 2, ...M \qquad \text{----------(2)}$$

$z_j = (f_1^{min}, f_2^{min}, ..., f_M^{min})$ *represents the ideal point.*

- For any given reference direction, the performance of a solution can be judged using two measures d_1 *and* d_2 *as depicted in Equation 3 and 4.*

- The first measure d_1 *is the Euclidean* distance between origin and the foot of the normal drawn from the solution to the reference direction, while the second measure d_2 *is the length of the normal.*

$$d_1 = w^T f_j'(x) \quad \text{----------(3)}$$

$$d_2 = \| f_j'(x) - f_j'(x) w^T w \| \quad \text{----------(4)}$$

For *j=1 to M* number of Objectives.

It is clear that a value of $d_2 = 0$ *ensures the solutions are perfectly aligned along the required reference direction* ensuring perfect diversity, while a smaller value of d_1 *indicates superior convergence.*

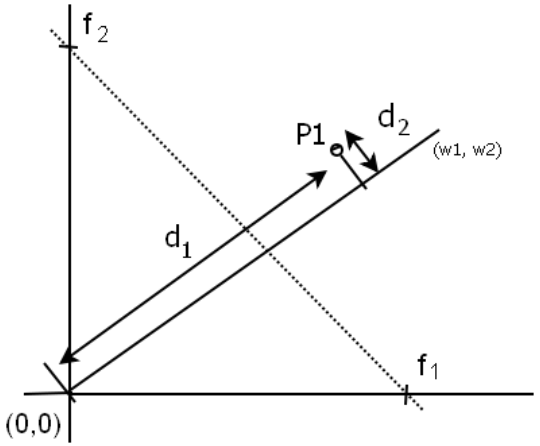


Fig. 6. Distance measures for a point p1 in two objectives

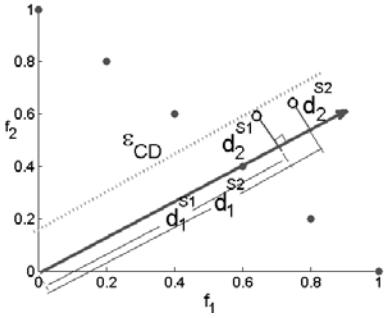
➢ **Mating Partner Selection**

- The mating partner for *Pi (where i is the index of the current* individual in a population) is selected using of the following rules i.e. rule 1: select a parent from the neighbourhood with a probability of τ and rule 2: select a random parent from the population with a probability of (1−*τ).*

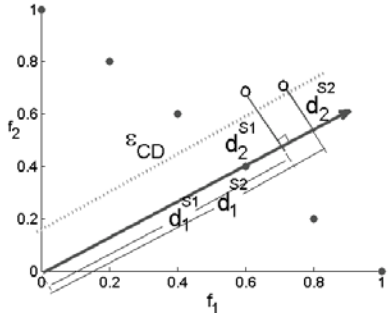
➢ Method of Recombination

- In the recombination process, two child solutions are generated using simulated binary crossover (SBX) operator and polynomial mutation. The first child is considered as an individual attempting to replace any parent in the population.

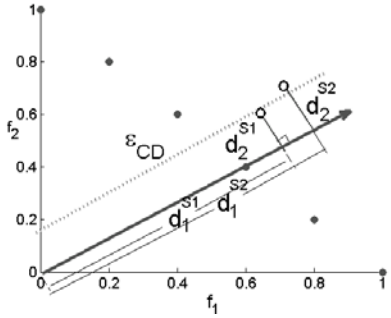
➢ Adaptive Epsilon Comparison to Manage the Balance between Convergence and Diversity



Case 1: Both the solutions have their d_2 *values less than εCD. One* with the smaller d_1 *is selected i.e.(s1)*



Case 2: Both the d_2 *values are more than εCD.* One with the lower d_2 *value is selected i.e.(s2).*



Case 3: One solution has its d_2 *value* more than *εCD and the other has its* d_2 *value less than εCD*

- The average deviation ε_{CD} *for* the population of solutions is computed using Equation 5

$$\varepsilon_{CD} = \frac{\sum_{i=1}^{W} d_2}{W} \qquad \text{----------(5)}$$

An Illustrative Example

- In order to observe the process of evolution, we computed the average performance of the population i.e. average of the d_1 *and* d_2 *values for the individuals for DTLZ1 (3* objectives)

- One can observe from Fig. 8, that the average d_2 *converges to near zero* (i.e. near perfect alignment to the reference directions) while the average d_1 *measure* stabilizes at around 0.8 in the normalized plane indicating convergence to the Pareto front

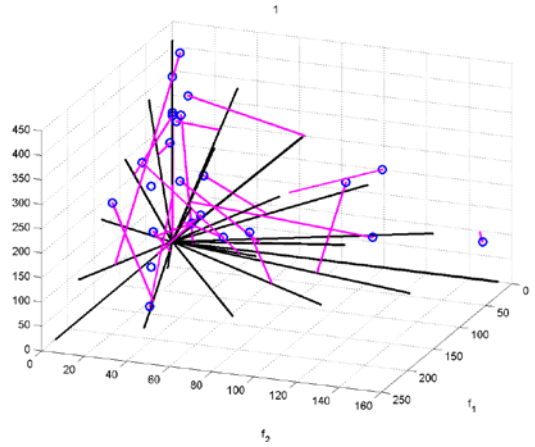


Fig:7. Evolving the best solutions with minimum d_1 and d_2 distances

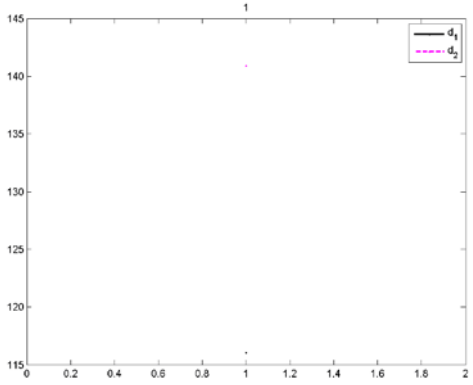


Fig:8. Converging the d_1 and d_2 measures over generations

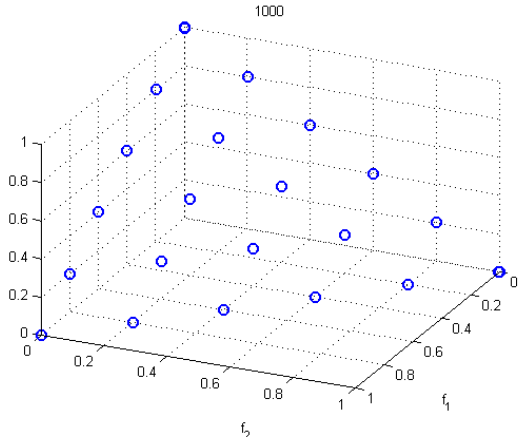


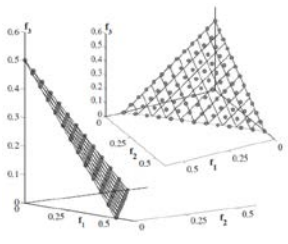
Fig:9. Final non-dominated solutions achieved for DTLZ1 problem

Experimental Results

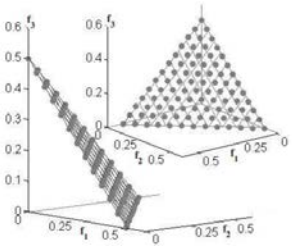
➢ Performance on Unconstrained DTLZ Problems

- In this comparison, we have reported the best, median and worst IGD results obtained using 20 independent runs for DTLZ1 and DTLZ2. The results are compared against M-NSGA-II and MOEA/D-PBI.

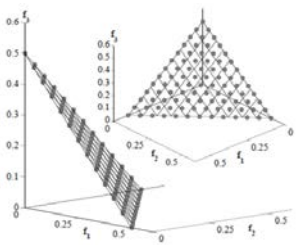
- One can observe that our algorithm obtained the best IGD values in 8 instances out of 10



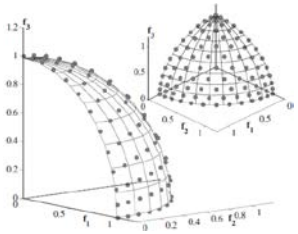
DBEA-Eps



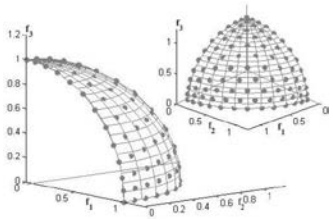
MOEA/D-PBI



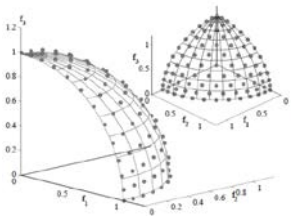
M-NSGA-II



DBEA-Eps



MOEA/D-PBI



M-NSGA-II

Test Problem	Obj.	MaxGen	Strategy	Best	Median	Worst
DTLZ1	3	400	DBEA-Eps	**8.771e-5**	9.521e-3	5.854e-1
			M-NSGA-II	4.880e-4	1.308e-3	4.880e-3
			MOEA/D-PBI	4.095e-4	**1.495e-3**	**4.743e-3**
DTLZ1	5	600	DBEA-Eps	**1.771e-5**	**2.183e-4**	3.782e-1
			M-NSGA-II	5.116e-4	9.799e-4	1.979e-3
			MOEA/D-PBI	3.179e-4	6.372e-4	**1.635e-3**
DTLZ1	8	750	DBEA-Eps	**4.387e-5**	**3.581e-4**	**1.981e-3**
			M-NSGA-II	2.044e-3	3.979e-3	8.721e-3
			MOEA/D-PBI	3.914e-3	6.106e-3	8.537e-3
DTLZ1	10	1000	DBEA-Eps	**7.691e-4**	**1.504e-3**	**2.700e-3**
			M-NSGA-II	2.215e-3	3.462e-3	6.869e-3
			MOEA/D-PBI	3.872e-3	5.073e-3	6.130e-3
DTLZ1	15	1500	DBEA-Eps	**1.696e-3**	**2.606e-3**	**2.686e-3**
			M-NSGA-II	2.649e-3	5.063e-3	1.123e-2
			MOEA/D-PBI	1.236e-2	1.431e-2	1.692e-2
DTLZ2	3	250	DBEA-Eps	2.040e-2	4.138e-2	6.417e-2
			M-NSGA-II	1.262e-3	1.357e-3	2.114e-3
			MOEA/D-PBI	**5.432e-4**	**6.406e-4**	**8.006e-4**
DTLZ2	5	350	DBEA-Eps	**1.199e-3**	3.024e-3	2.272e-2
			M-NSGA-II	4.254e-3	4.982e-3	5.862e-3
			MOEA/D-PBI	1.219e-3	**1.437e-3**	**1.727e-3**
DTLZ2	8	500	DBEA-Eps	**1.172e-3**	**2.899e-3**	6.915e-3
			M-NSGA-II	1.371e-2	1.571e-2	1.811e-2
			MOEA/D-PBI	3.097e-3	3.763e-3	**5.198e-3**
DTLZ2	10	750	DBEA-Eps	3.656e-3	3.657e-3	3.657e-3
			M-NSGA-II	1.350e-2	1.528e-2	1.697e-2
			MOEA/D-PBI	**2.474e-3**	**2.778e-3**	**3.235e-3**
DTLZ2	15	1000	DBEA-Eps	**5.160e-3**	**5.960e-3**	**5.960e-3**
			M-NSGA-II	1.360e-2	1.726e-3	2.114e-2
			MOEA/D-PBI	5.254e-3	6.005e-3	9.409e-3

- Results using systematic sampling for DTLZ1 and DTLZ2 problems for all algorithms

➢ Constrained Engineering Design Problems

▪ Car Side Impact Problem

• The problem aims to minimize the weight of a car, the pubic force experienced by a passenger and the average velocity of the V-Pillar responsible for bearing the impact load subject to the constraints involving limiting values of abdomen load, pubic force, velocity of V-Pillar, rib deflection etc

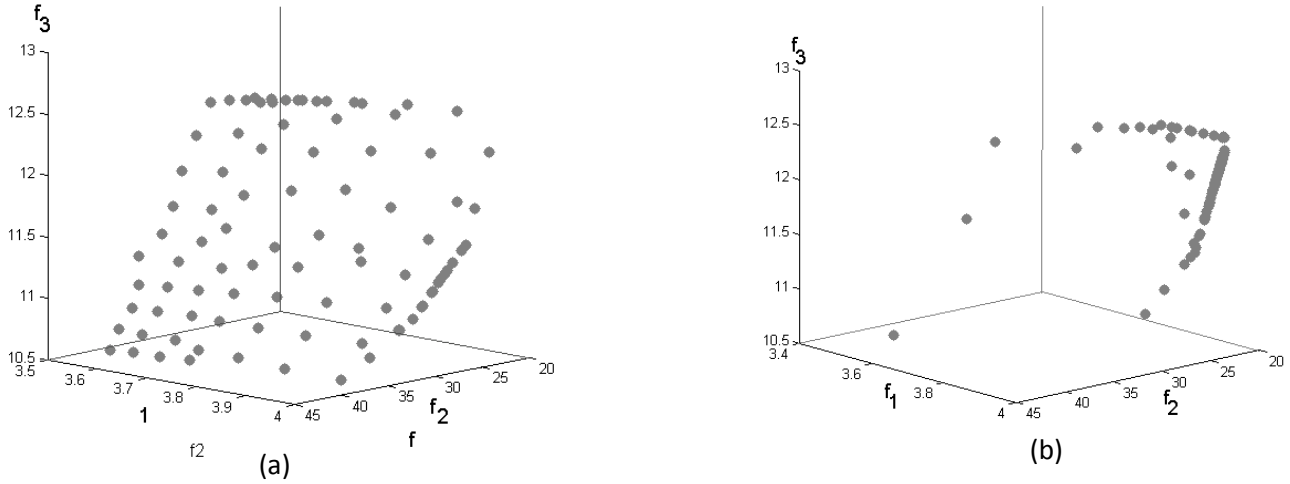


Fig. 10. Solutions obtained using (a) DBEA-Eps (b) MOEA/D-PBI on three-objective car side impact problem

• The algorithms are run for 500 generations and the final non-dominated front is shown in Figure. It is important to note that the results of MOEA/D-PBI is derived without scaling which could be a reason among others for poor performance.

- **Water Resource Management Problem**

• This is a five objective problem having seven constraints taken from the literature. The parallel coordinate plot generate using our proposed algorithm (DBEA-Eps) is presented in Fig. 11.

• The best IGD value across 20 runs is 3.*29e−2 and the IGD is computed* using the reference set of 2429 solutions

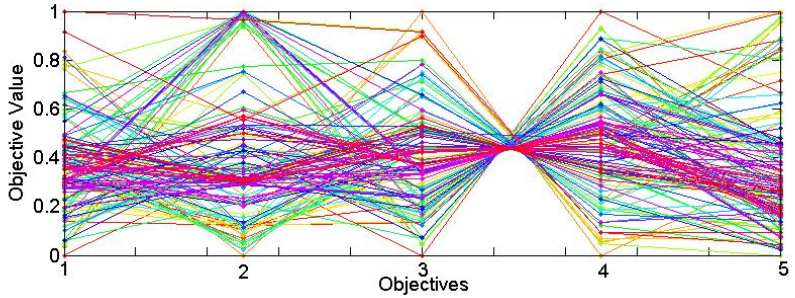


Fig. 11. Solutions obtained using DBEA-Eps on five-objective water problem

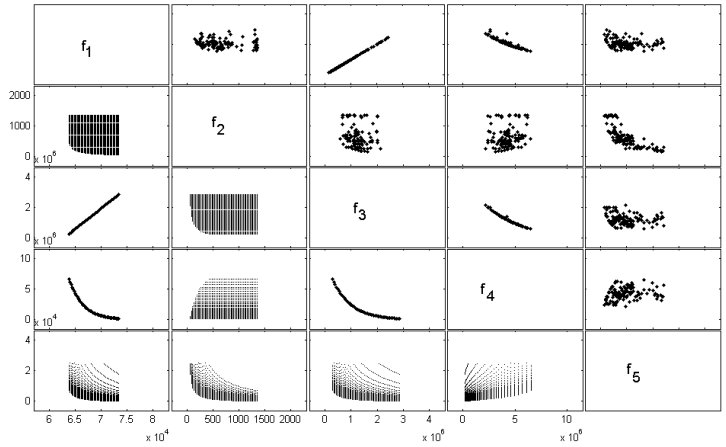


Fig. 12. A scatter plotmatrix showing DBEA-Eps (top-right plots) vis-a-vis the known reference set of 2429 solutions (bottom-left plots)

• A scatter plot-matrix is presented. The results from the DBEA-Eps are shown in the top-right plots vis-a-vis the known reference set of 2429 solutions (shown in bottom-left plots).

Experimental Results

- **General Aviation Aircraft (GAA) Design Problem**

- This problem was first introduced by Simpson. The problem involves 9 design variables i.e. cruise speed, aspect ratio, sweep angle, propeller diameter, wing loading, engine activity factor, seat width, tail length/ diameter ratio and taper ratio and the aim is to minimize the takeoff noise, empty weight, direct operating cost, ride roughness, fuel weight, purchase price, product family dissimilarity and maximize the flight range, lift/ drag ratio and cruise speed

- In this example, we have used 100 reference points and the population was allowed to evolve over 5000 generations. A reference set of 412 non-dominated solutions obtained from ε-MOEA and Borg-MOEA are used to compute the IGD metric

Table 2. Performance metric value of product family design problem using 50 independent runs

Algorithm	Function Evaluation	IGD			
		Best	Mean	Worst	Std
DBEA-Eps	50,000	0.62070	0.80123	0.82430	0.09210
ε-MOEA		0.98312	0.99123	0.99678	0.10312
Borg-MOEA		0.98211	0.99113	0.99337	0.02321
MOEA/D		0.99117	0.99587	0.99723	0.02145
ε-NSGA-II		0.98571	0.98872	0.99131	0.72123

- The performance of the algorithms is compared using the hyper-volume in Table 2 and IGD in Table 3. One can observe that the proposed algorithm performs marginally better than others for this problem.

Table 3. Performance metric value of product family design problem using 50 independent runs

Algorithm	Function Evaluation	Hypervolume			
		Best	Mean	Worst	Std
DBEA-Eps	50,000	0.02899	0.01715	0.00689	0.04561
ε-MOEA		0.02032	0.01032	0.00259	0.04125
Borg-MOEA		0.02245	0.01013	0.00424	0.02327
MOEA/D		0.00092	0.00087	0.00045	0.00145
ε-NSGA-II		0.01636	0.01005	0.00236	0.05232

Conclusion and Future Work

- In this paper, a decomposition based evolutionary algorithm with adaptive epsilon comparison is introduced to solve unconstrained and constrained many objective optimization problems.

- The approach utilizes reference directions to guide the search, wherein the reference directions are generated using a systematic sampling scheme.

- In an attempt to alleviate the problems associated with scalarization (commonly encountered in the context of reference direction based methods), the balance between diversity and convergence is maintained using an adaptive epsilon comparison.

- In order to deal with constraints, an epsilon level comparison is used which is known to be more effective than methods employing *feasibility first principles.*

- Three constrained engineering design optimization problems with three to seven constraints (car side impact, water resource management and a general aviation aircraft design problem) have been solved to illustrate the performance of the proposed algorithm.

- The preliminary results indicate that the proposed algorithm is able to deal with unconstrained and constrained many objective optimization problems better or at par with existing state of the art algorithms such as M-NSGA-II and MOEA/D-PBI.

Further Improvements

- Due to improper balance between d_1 and d_2 with the epsilon level Comparison, the solutions can converge to a local optima

- In the steady state form, if a child solution is non-dominated with respect to the reference set, it enters the reference set via a replacement. The child solution competes with all solutions in the reference set in a random order until it makes a successful replacement or have competed with all individuals in the reference set.

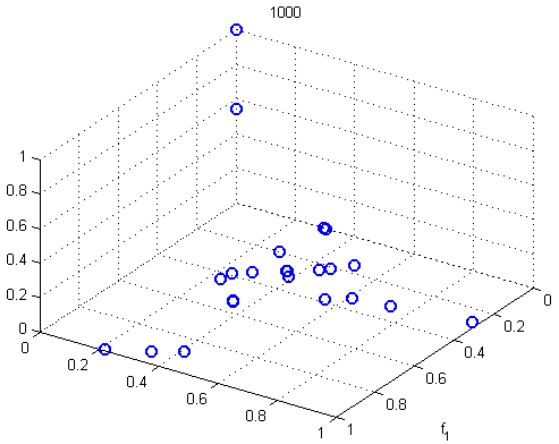


Fig: 13. Locally converged solutions for problem DTLZ1 on 3-objective

- If we denote the distances as *{d_{1r}, d_{2r} } for a r^{th} solution in the reference set and {d_{1c}, d_{2c} } denotes the distances for the child solution along r^{th}* reference direction, the replacement rule is as Algorithm 2.

Algorithm 2 Comparison of two solutions

```
if d2c = d2r then
    if d1c < d1r then
        Child solution wins
    end if
else
    if d2c < d2r then
        Child solution wins
    end if
end if
```

Improved solutions for the same problem DTLZ1 on 3 objectives with same number of function evaluations

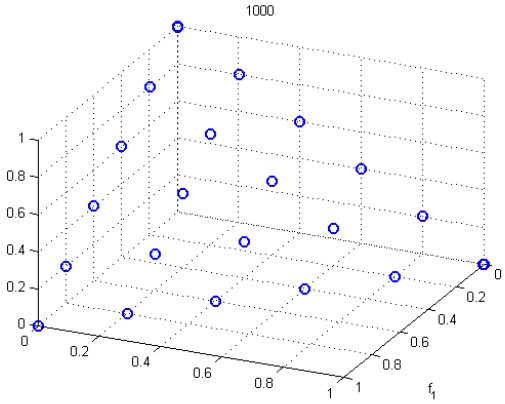


Fig: 14. Final Pareto solutions for problem DTLZ1 on 3-objective with modified comparison

Further Improvements

- The practicality of solving many-objective optimization is often questioned owing to the fact that even if the whole direction vector is available, there are no suitable means for the solutions to align properly, and hence it is difficult to choose a preferred solution out of the innumerable Pareto optimal solutions. Therefore, the solutions can converge to a local optima.

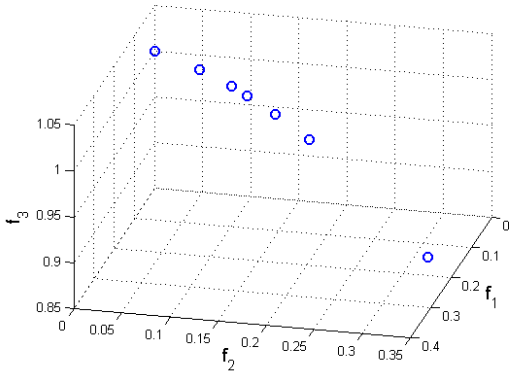


Fig: 15. Locally converged solutions for problem DTLZ5(2,5)

- In order to maintain the corner solutions, the corner solutions of a problem should be preserved externally.

Improved Pareto front for DTLZ5(5,2) problem by preserving the corner solutions externally

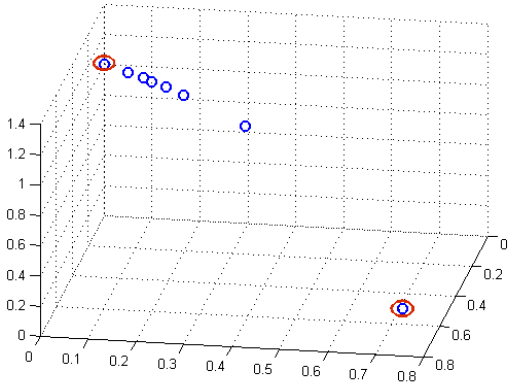


Fig: 16. Final Pareto solutions for problem DTLZ5(2,5) preserving corner solutions

- Sample Population for a Three- Objective Problem

Solution ID	f_1	f_2	f_3
1	0.0617	0.1561	0.1173
2	0.3924	0.5660	0.0336
3	0.5446	0.0183	0.4089
4	0.6359	0.2619	0.0731
5	0.0365	0.7365	0.6474
6	0.2322	0.4008	0.0357
7	0.2440	0.3225	0.1113
8	0.6014	0.0876	0.1886
9	0.9205	0.1960	0.1153
10	0.7453	0.0277	0.2315
11	0.1123	0.0914	0.3017
12	0.0551	0.5851	0.7805

Rank	f_1	f_2	f_3	$f_2^2 + f_3^2$	$f_1^2 + f_3^2$	$f_1^2 + f_2^2$
1					1	
2	12	10	6	8		1
3	1	8	4	9	7	7
4	11	11	7	10	11	6
5	6	1	9	4	2	3
6	7	9	1	11	8	12
7	2	4	8	7	4	8
8	3	7	10	6	5	4
9	8	6	11	3	3	2
10	4	2	3	2	10	5
11	10	12	5	12	12	10
12	9	5	12	5	9	9

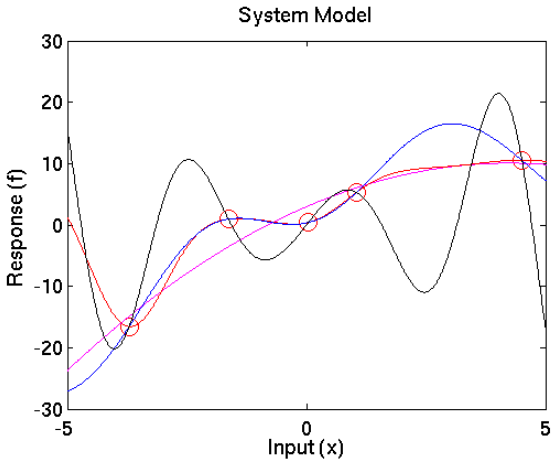
Pareto Corner sort example

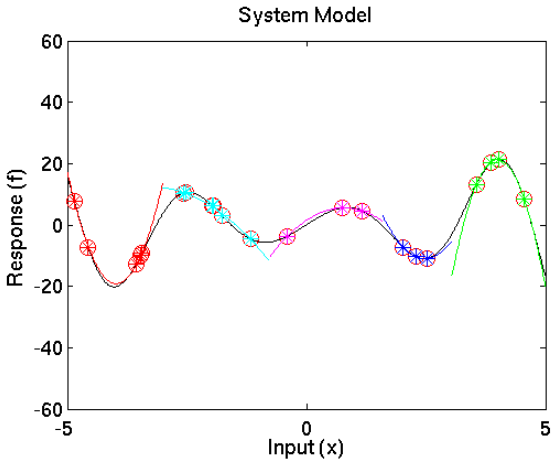
Managing Computationally Expensive Analysis: Surrogate Assisted Optimization

Surrogate Assisted Optimization

- Evaluation of a solution is often a computationally expensive process.
- Use approximations or surrogates within an optimization loop.
- Different type of surrogates have different behavior. Choice of the most appropriate one is non-trivial.
- A function can be better approximated by a type of surrogate over another.
- The same function in different regions can be better approximated using different surrogates.

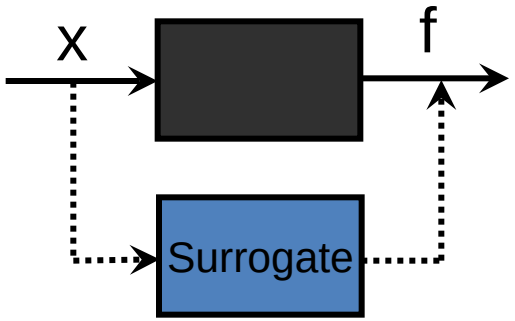
- Our approach is to use: Multiple Spatially distributed Surrogates of Multiple Types.
- RSM, RBF, MLP, Kriging Models coexist and are attempted on each objective and constraint function.
- A minimum prediction error threshold is used to invoke any surrogate call, else an actual evaluation is invoked.
- Solutions with inadequate neighborhood sampling using actual analysis always use actual analysis.





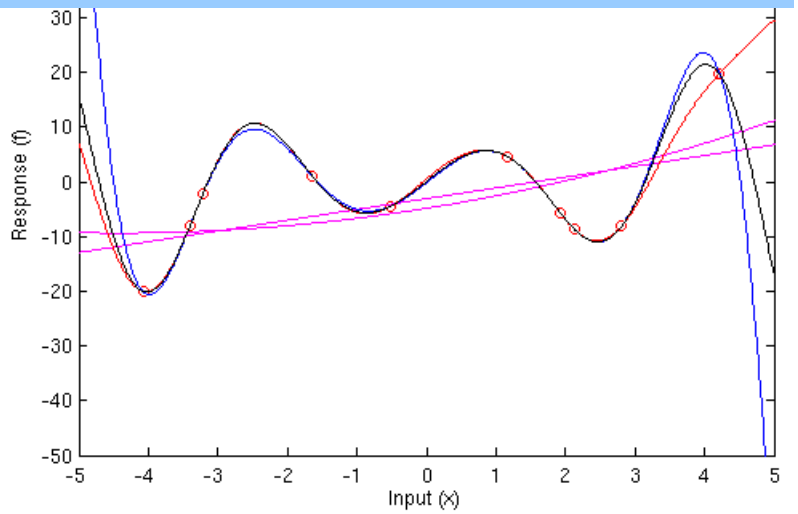
Surrogate Assisted Optimization

- Consider another system
  - It's an **unknown** function
  - Experiments



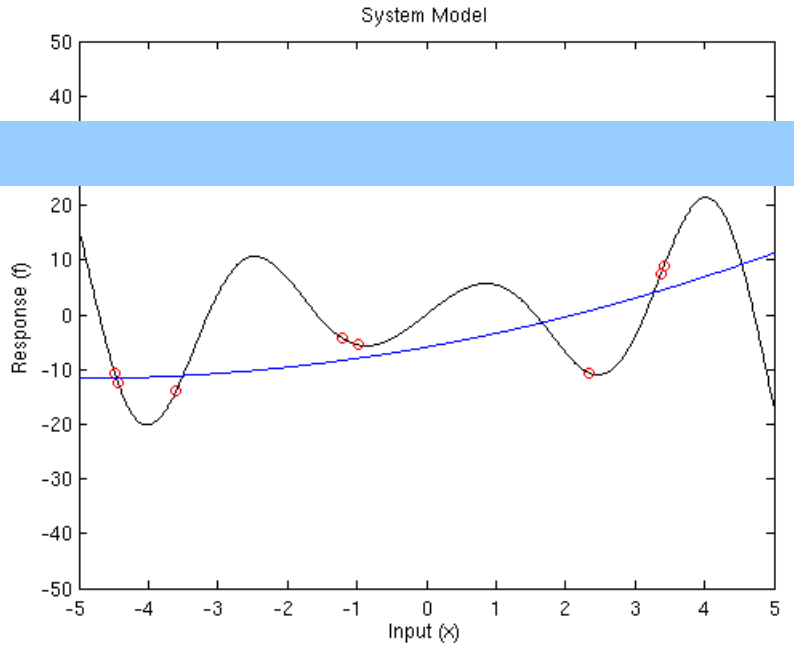
- Approximation Models
  - Linear
  - Kriging

- Surrogates are cheaper than actual evaluations
- Choice of surrogate dictates approximation accuracy



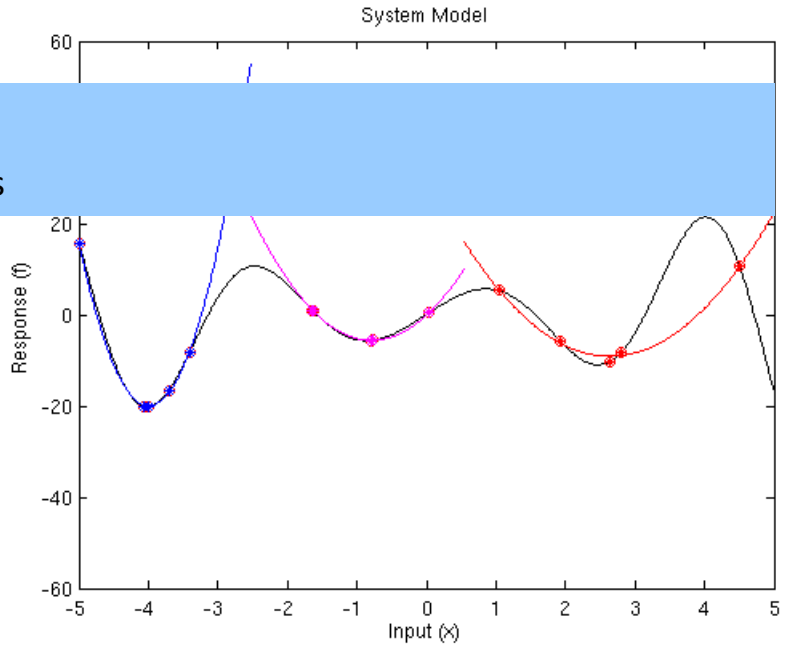
Single Surrogates

- Consider earlier system
  - What is the smallest value of f?
- Build Surrogate
  - Sampling
  - Quadratic Model
- Evolutionary Steps
  - Single global surrogate has difficulty approximating
  - Find smallest value
  - Verify using actual model
- Re-train Surrogate
- Repeat …



Multiple Surrogates

- Using the same system
  - What is the smallest value of f?
- Build Many Surrogates
  - Sampling
  - Grouping (Clustering)
  - • Multiple surrogates approximate the system better
  - • Performance depends on number of groups/clusters
  - Find smallest values
  - Verify using actual model
- Re-group and Re-train
- Repeat …



Adaptive Multiple Surrogates

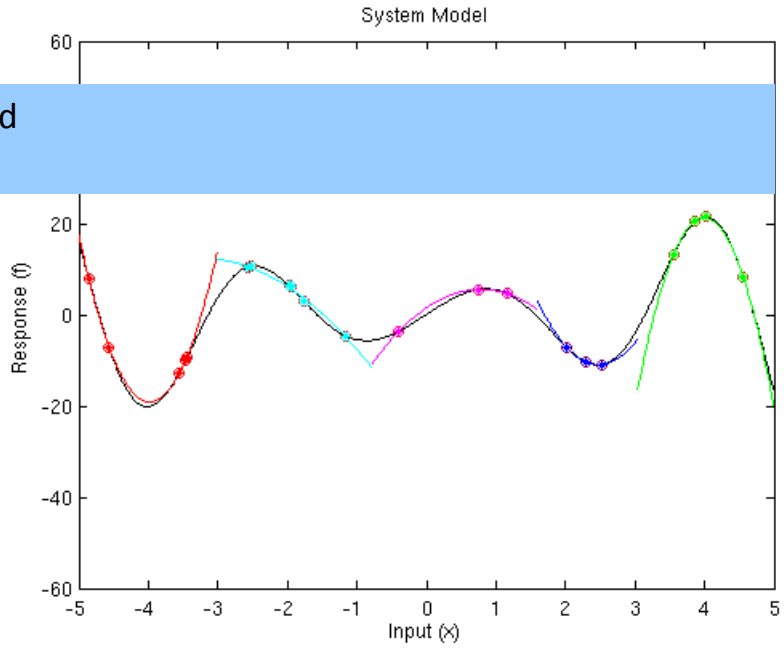
- Last time the same system

- Sampling
- Building Many Surrogates
  - Adaptive clustering picks the number of clusters (and surrogates) which approximate the system best!

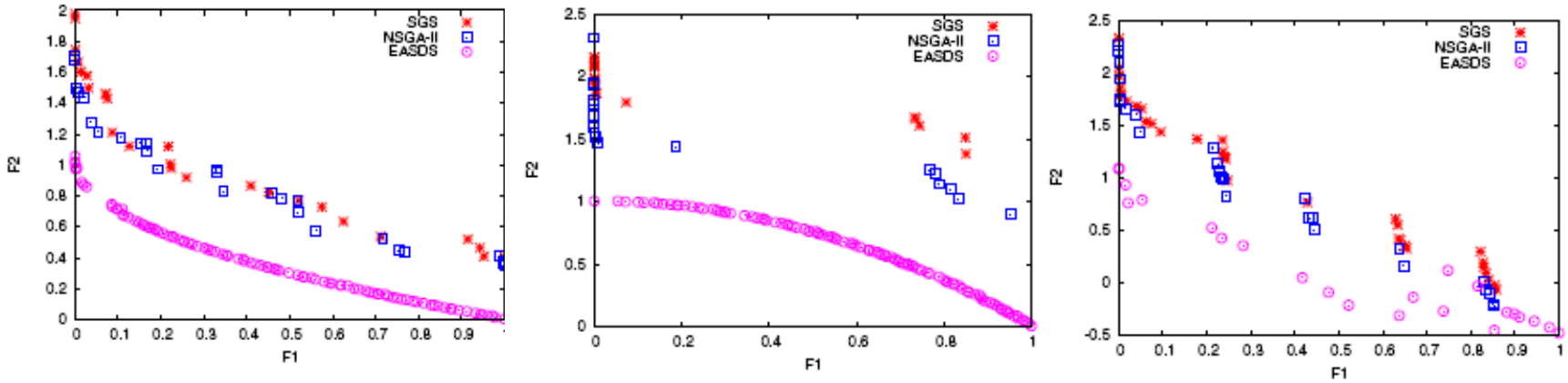
  4 Clusters

  5 Clusters
- Choose best Clustering

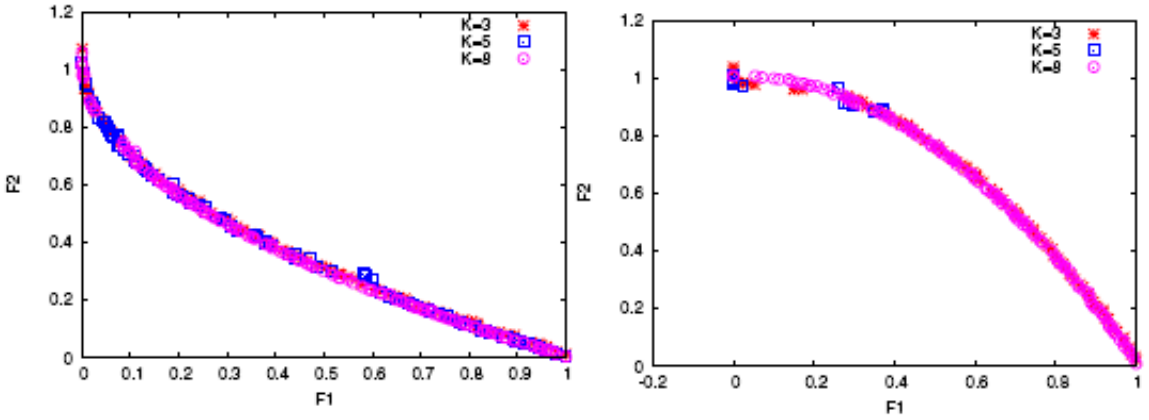
  Every evolutionary step



Performance



ZDT1, ZDT2 and ZDT3: Same number of function evaluations, Same starting population, 2100 evaluations.



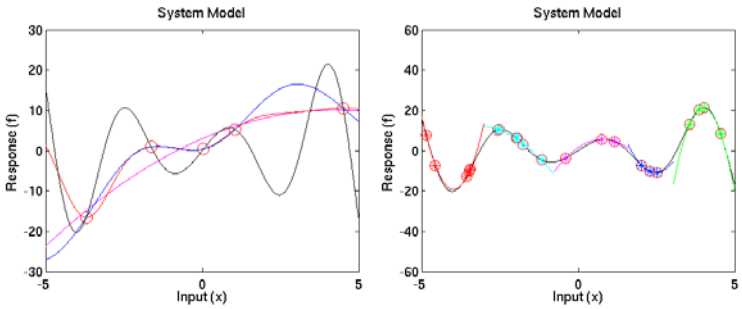
ZDT1 and ZDT2: Effect of Number of Clusters

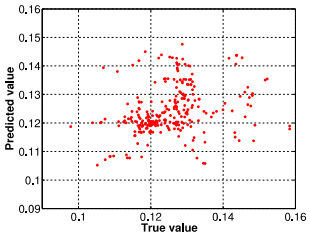
Isaacs, A. , Ray, T. , and Smith, W. , "Multiobjective design optimization using multiple adaptive spatially distributed surrogates," International Journal of Product Development, vol. 9, no. 1-3, pp. 188–217, 2009.

Extreme Surrogate Assisted / Multifidelity Optimization Capability

MDO RESEARCH GROUP
Multidisciplinary Design Optimization

- Given a set of data, we offer a highly sophisticated framework for generating approximations(surrogates).
- Some functions are better approximated by a type of surrogate over another.
- The same function in different regions can be better approximated using different surrogates.
- RSM, RBF, MLP, Kriging Models coexist and are attempted on each function.
- **Our approach is to use: Multiple Spatially distributed Surrogates of Multiple Types.**





Isaacs, A. , Ray, T. , and Smith, W. , "Multiobjective design optimization using multiple adaptive spatially distributed surrogates," International Journal of Product Development, vol. 9, no. 1-3, pp. 188–217, 2009.

Multifidelity-Optimization

samples through expensive and combined approaches for the toy submarine design problem

Using 12 accurate CFD analysis, the performance of a surrogate model is evaluated on 277 unseen design instances (Fig 1. (red). Using 12 accurate and 277 inaccurate CFD estimates, the performance of the combined model is presented in Fig1. (blue). Accurate CFD refers to one where the simulation is run for 100 iterations, while inaccurate refers to 50 iterations.

We can boost model performance using data from different sources, different quantities and even via hypothetical quantities. Hence generation of combined models using data from multiple published sources or combinations of experimental and CFD results is now possible.

UNSW THE UNIVERSITY OF NEW SOUTH WALES

Need for Robust Solutions:

Robust Optimization

Robust Optimization

$$\left.\begin{array}{l}\text{Minimize}(f_1(x), f_2(x), \ldots f_M(x)), \\ \text{subject to } \dfrac{\|f^p(x) - f(x)\|}{\|f(x)\|} \leq \eta, \\ RCV(x) \geq 0, x \in S\end{array}\right\}$$

$$\text{RCV(x)} = \Sigma_{x^* \in \delta_x} CV(x^*) + GCV(x^*)$$

Algorithm 1 IDEAR

Require: P {Population Size}
Require: $N_{\max} > 1$ {Number of max. function evaluations}
Require: $0 < \alpha < 1$ {Proportion of Infeasible Solutions}
Require: H, δ_x, and η
1: Initialize pop, pop_p
2: Eval. pop_p
3: **repeat**
4: Evolve pop_p to get $childpop$, pop_c
5: Eval. *robustness constraints* of pop_c using $H = H/2$
6: $pop_n \Leftarrow ND\_select(pop_p + pop_c)$
7: Eval. *robustness constraints* of pop_c solutions in pop_n
8: $pop_n \Leftarrow ND\_select(pop_n + pop_c)$
9: $pop_p = pop_n$
10: $N_{\text{cur}} \Leftarrow$ Update functioneval. count
11: **until** $N_{\text{cur}} > N_{\max}$

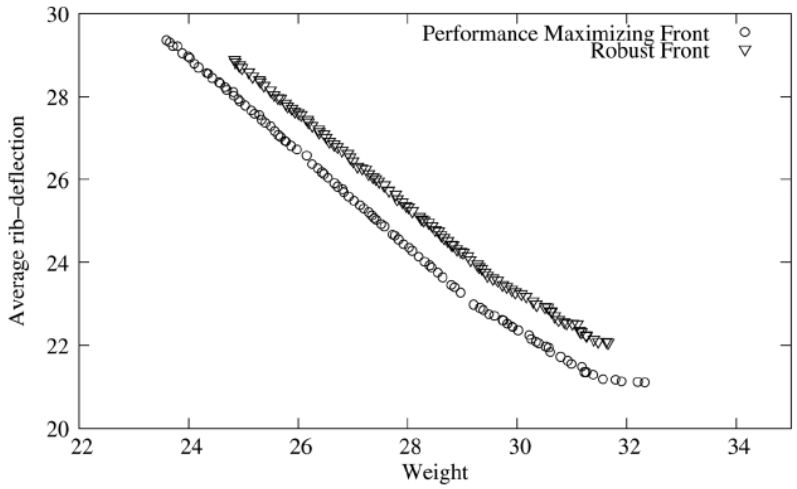


Fig. 8 Robust optimal solutions of the car side impact problem as compared to the performance maximizing solutions

Saha, A. and Ray, T.(2011), Practical Robust Design Optimization using Evolutionary Algorithms, *ASME Journal of Mechanical Design*, Vol. 133, October 2011, pp. 101012-1 - 101012-19.

Performance of Robust Design Optimization Methodology

Table 5: Performance criteria of the existing USS Dallas RC toy, optimum and robust submarines

Vehicle particulars	USS Dallas RC	Optimum	Robust
Nose length	45 mm	49 mm	62 mm
Parallel middle body length	210 mm	231 mm	307 mm
Tail length	95 mm	80 mm	78 mm
Length overall	350 mm	360 mm	447 mm
Maximum diameter	60 mm	58 mm	58 mm
Length to diameter ratio	5.8	6.2	**7.7**
Maximum dimension of the inner square	39.6 mm	38 mm	38 mm
Wetted surface area	0.082385 m^2	0.082624 m^2	0.102018 m^2
Displacement volume	0.000437 m^3	0.000433 m^3	0.000598 m^3
Mass of the displaced water	437 g	433 g	598 g
Total mass of the vehicle	430 g	428 g	475 g
Length of the first lever arm	45 mm	48 mm	**71 mm**
Length of the second lever arm	90 mm	90 mm	**143 mm**
X-coordinate of CG	-0.981462 mm	-0.982488 mm	-0.883385 mm
Y-coordinate of CG	-0.210313 mm	-0.210533 mm	-0.189297 mm
Z-coordinate of CG	167.083 mm	172.734 mm	203.566 mm
X-coordinate of CB	0	0	0
Y-coordinate of CB	0	0	0
Z-coordinate of CB	163.375 mm	169.906 mm	205.737 mm
Longitudinal distance between CB and CG	3.705 mm	2.828 mm	**2.171 mm**
Nominal speed	0.5 m/s	0.5 m/s	0.5 m/s
Drag (VT method)	0.0792024 N	**0.0789568 N**	0.090821 N
Drag (G&J method)	0.0800956 N	**0.0798317 N**	0.0913319 N
Drag (MIT method)	0.0825858 N	**0.0821771 N**	0.0924765 N

Toy Submarine Design Problem

Alam, K. , Ray, T. and Anavatti, S. ,"A new robust design optimization approach for unmanned underwater vehicle design," Journal of Engineering for the Maritime Environment, In Press, (Accepted 11/2011).

Uncovering Design Rules

Uncovering Design Rules

Mining the data generated out of an optimization process can lead to the identification of useful design rules.

$$
\begin{aligned}
\text{Minimize} \quad & f(L_I, B_I, T_I) = L_I^{0.395} B_I^{0.89} T_I - 36.04 \\
\text{Subject to} \quad & g_1 : \Delta_I \geq \Delta_B \\
& g_2 : GM_I \geq GM_B \\
& g_3 : 3.07 \leq L_I / V_I^{1/3} \leq 12.4 \\
& g_4 : 3.7^o \leq Ie_I \leq 28.6^o \\
& g_5 : 2.52 \leq L_I / B_I \leq 18.28 \\
& g_6 : 1.7 \leq B_I / T_I \leq 9.8 \\
\text{Variable bounds} \quad & 29.93m \leq L_I \leq 35.93m \quad (L_B = 32.93m) \\
& 6.51m \leq B_I \leq 8.51m \quad (B_B = 7.51m) \\
& 1.50m \leq T_I \leq 1.70m \quad (T_B = 1.59m)
\end{aligned}
$$

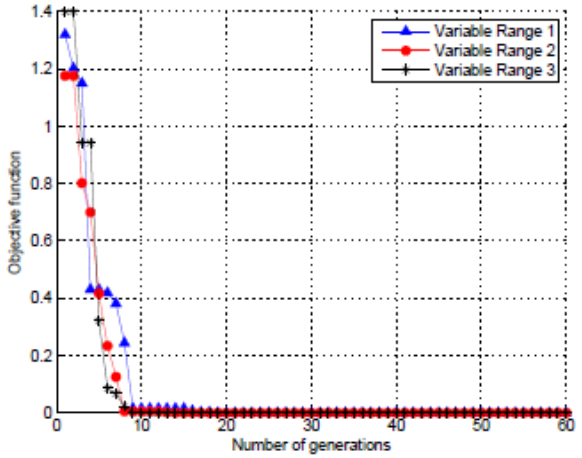


FIGURE 6.5: PROGRESS PLOT OF THREE SINGLE-OBJECTIVE OPTIMIZATION USING DIFFERENT VARIABLE BOUND RANGES

Mohamad, A.F.A. , **Ray, T.** , and Smith, W. , "Uncovering secrets behind low resistance planing craft hull forms through optimization," Engineering Optimization, vol. 43, no. 11, November, pp. 1161–1173, 2011.

Mohamad, A.F.A. , **Ray, T.** , and Smith, W. , "A hydrodynamic preliminary design optimization framework for high speed planing craft," Journal of Ship Research, vol. 56, No. 1, pp. 35–47, 2012.

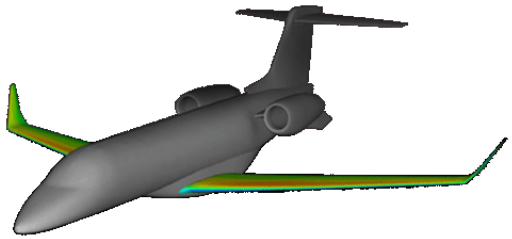
Application Snapshot: Shape Representation and Optimization

Basics

- In shape optimization, a designer requires to adopt a shape representation scheme.
- The variables of such a scheme should be as few as possible, yet it should reflect changes to the function or constraints of the optimization problem.
- Common variables include control points of Bezier Curves, Bsplines, NURBS or special encoding schemes such as those used for NACA airfoil shapes or PARSEC for representation of airfoil shapes.
- Typically with such shapes, computationally expensive analysis is invoked, i.e. CFD or FEM analysis is usually conducted on those shapes to find their worth.
- The underlying optimization problem thus is computationally expensive and this extreme care should be taken to ensure the variables are just enough to provide the designer with the flexibility.

Shape Representation

- Shape representation and optimization is a key element in any product design process.
- Shape representation schemes are required for the generation of shapes which in turn facilitates the design of functional articles.
- Aerofoils, converging-diverging nozzles, ship-hulls, medical prosthesis, structural elements and many other functional articles require shape generation and modification to achieve the desired performance.



Aircraft wing

Ship hull

Prosthesis

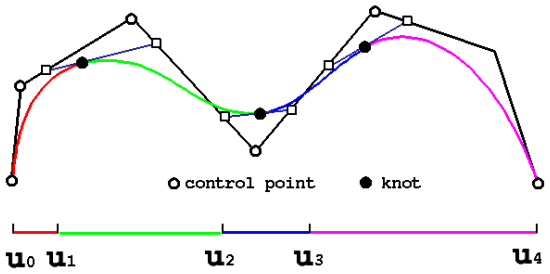
- B-spline Curve
  - Non-global behaviour.
  - Greater flexibility.
  - Degree of the curve does not influenced by control points.
  - Strong convex-hull property.
  - Variation diminishing and affine invariance properties.

A B-spline curve can be defined as [Cox, 1971]:

$$P(t) = \sum_{i=1}^{n+1} B_i N_{i,k}(t); t_{\min} \le t < t_{\max}; 2 \le k \le n+1$$

The basis functions $N_{i,k}(t)$ are defined by the Cox-de Boor's recursion formula:

$$N_{i,k}(t) = \frac{(t - x_i)N_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)N_{i+1,k+1}(t)}{x_{i+k} - x_{i+1}}$$



Shape Matching Metrics

- Euclidean Distance
  - Distance between two corresponding points of two shapes.

$$L_j(P, Q) = \left(\sum_{i=0}^{m} \left|p_i - q_i\right|^j\right)^{1/j}$$

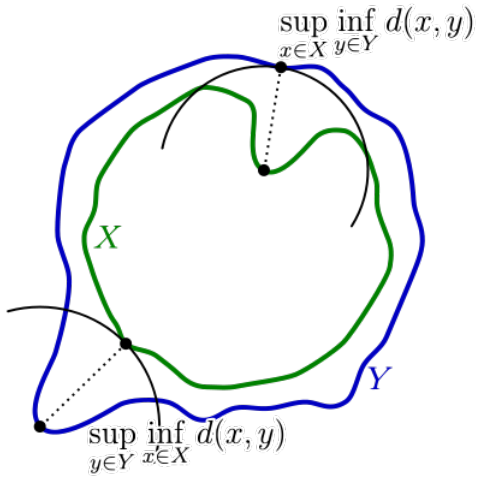
  - for j = 2, this yields the Euclidean distance [Remco, 2001].

- Hausdorff Distance
  - Classical non correspondence based shape matching method.
  - For two finite point sets *P* and *Q* the Hausdorff distance between
  - *P and Q* is defined as [Huttenlocher, 1990]:

$$H(P, Q) = \max\{h(P, Q), h(P, Q)\}$$

Where,

$$h(P, Q) = \max_{p \in P} \min_{q \in Q} \|p - q\|$$



2D Shape Representation and Optimization

Smart Repair

Boundary representation using B-Splines.

Novel Repair Method that ensures generation of valid geometries.

Our Proposed Repair Approach: A convex-hull is generated using a set of control points (Fig.2). Thereafter, the points (lying inside the convex-hull) nearest to their adjacent edges are inserted to generate the non-intersecting control polygon net (Fig.3).

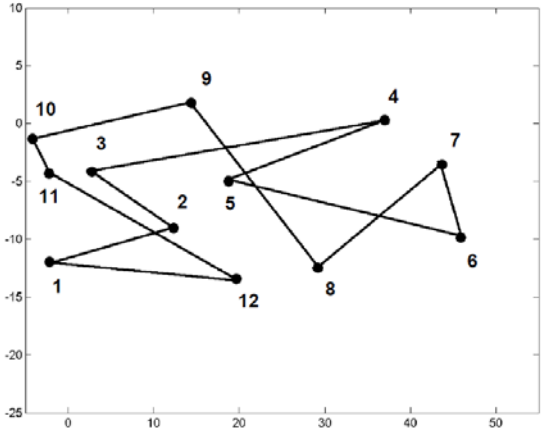


Fig.1 Position of initial control points without repair

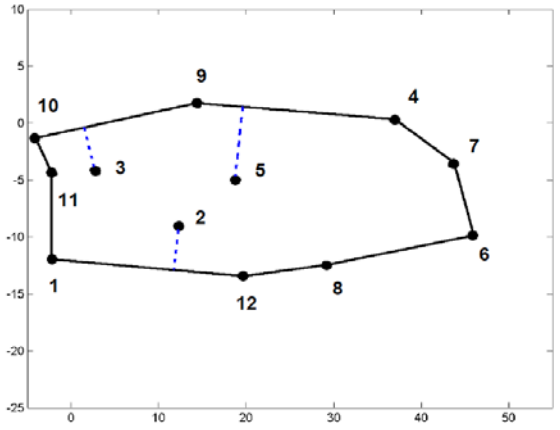


Fig .2 Convex-hull formation with initial control points

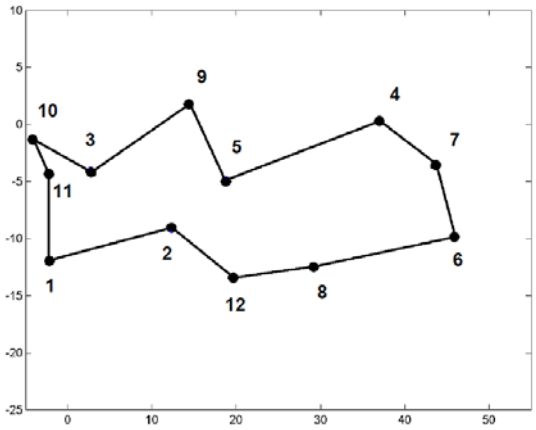


Fig.3 Position of initial control points after repair

Khan, M., Mohamad, A.F.A., Isaacs, A., and Ray, T.(2011), A novel evolutionary approach for 2D shape matching based on B-spline modeling, *IEEE Congress on Evolutionary Computation, CEC-2011*.

Khan, M., Mohamad, A.F.A., Isaacs, A., and Ray, T., A smart repair embedded memetic algorithm for 2D shape matching problems, *Engineering Optimization*, Accepted 29/09/2011.

Flapping Wing Geometry

- Wing geometry is one of the most important factors that affects the performance of a flapping wing.
- In order to gain an in-depth understanding of flapping flight with an aim to identify optimal wing shapes, there is a need for an universal and flexible shape representation scheme that is amenable to optimization.

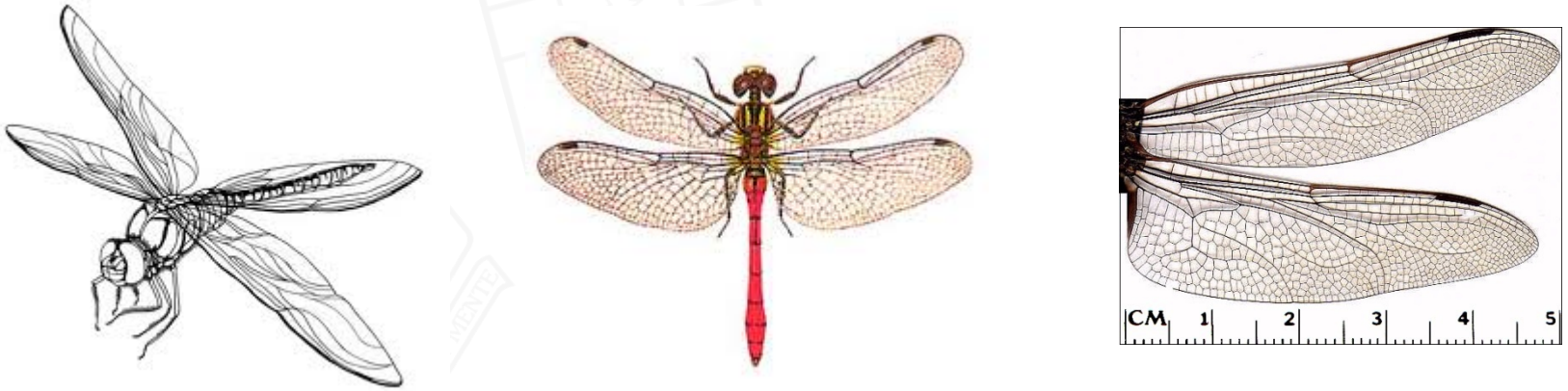


Fig.4 Flapping wing geometry of dragonfly

Khan, M. and Ray, T. , "Shape representation and a morphing scheme to support flapping wing research ," International Conference on Pattern Recognition Applications and Methods, ICPRAM-2012.

Flapping Wing Species

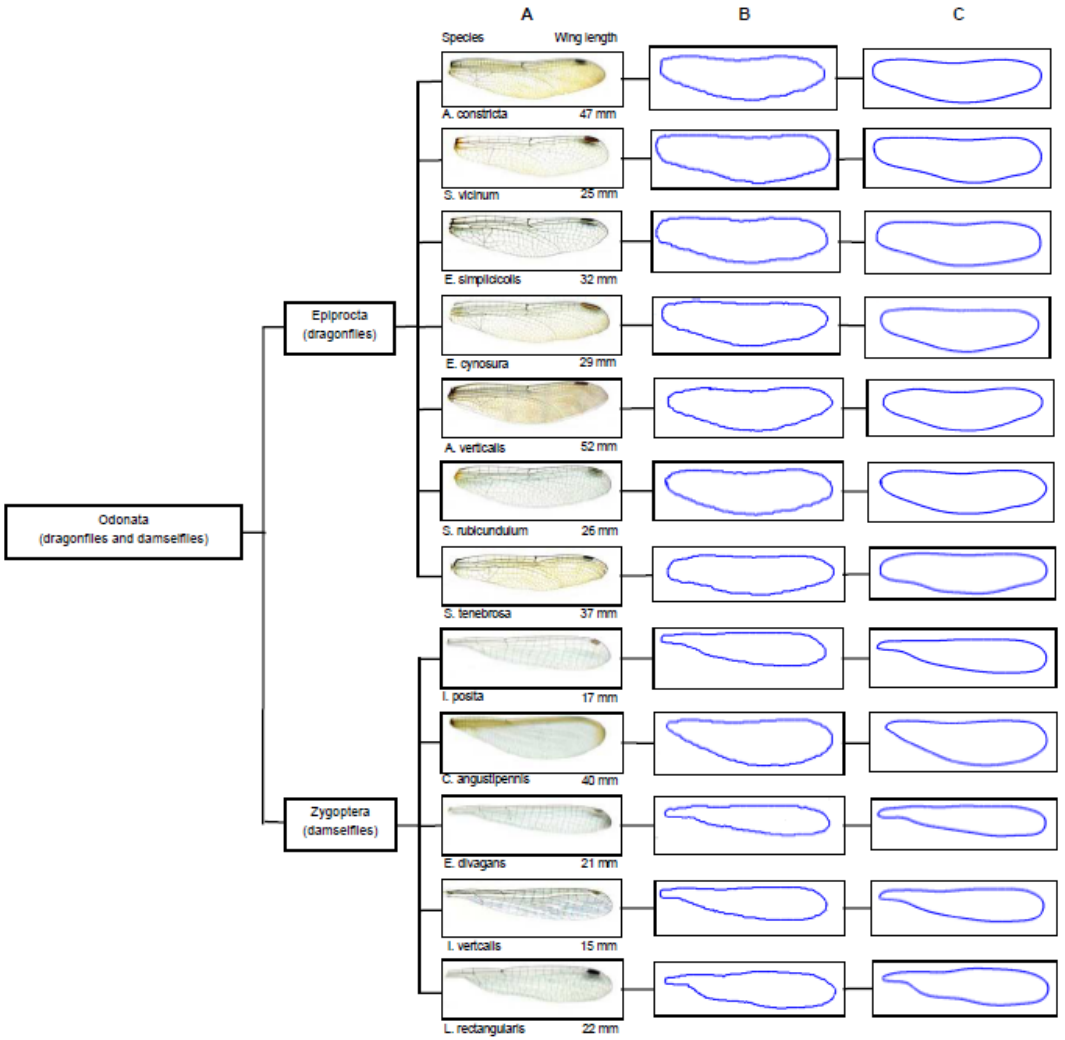


Fig.5 Dragonfly and Damselfly wing species and digitally extracted shapes' boundaries.

	Species	Length a, mm	Width b, mm	A.ratio a/b
Dragonfly	A. constricta	47	11.92	3.94:1
	S. vicinum	25	6.64	3.77:1
	E. simplicicolis	32	8.21	3.89:1
	E. cynosura	29	7.64	3.79:1
	A. verticalis	52	13.69	3.79:1
	S. rubicundulum	26	6.69	3.88:1
	S. tenebrosa	37	8.54	4.33:1
Damselfly	I. posita	17	3.36	5.06:1
	C. angustipennis	40	10.76	3.71:1
	E. divagans	21	3.59	5.84:1
	I. vertcalis	15	2.88	5.20:1
	L. rectangularis	22	4.68	4.70:1

Fig.6 Dragonfly and Damselfly wing dimensions with their aspect ratios

Performance of Smart Repair Strategies 2D

- The optimum design of the 2D Damselfly wing (contains 516 points in x and y coordinates) after 5,000 function evaluations.

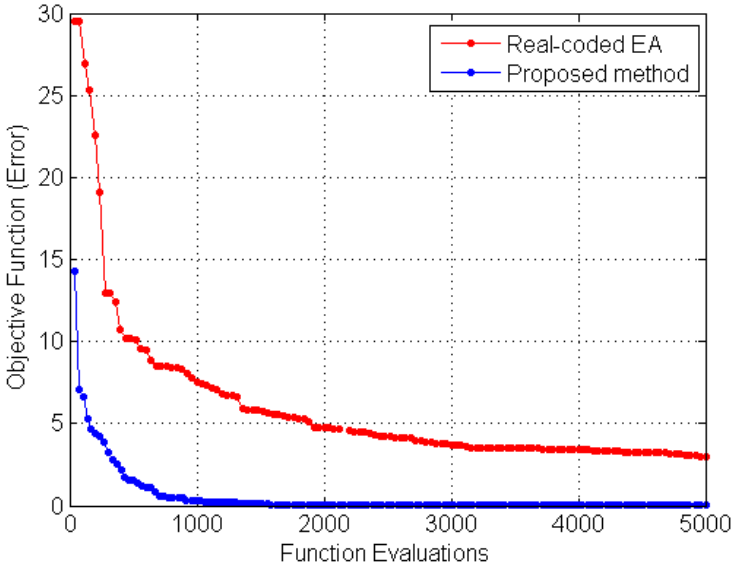
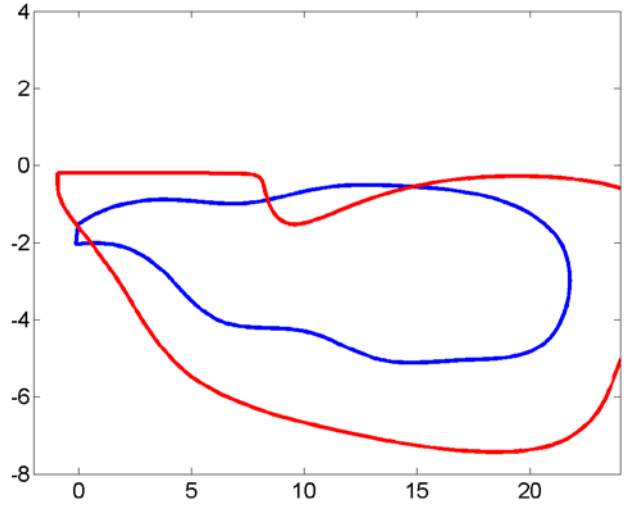


Fig.7 Progress plot of the best design for single objective matching error minimization



Ani.1 Evolution of Damselfly wing (red) towards the target Damselfly wing (blue)

Method	Error Measurement	Best	Worst	Mean	Median	Std.
Real-coded EA	Max (Eucli,HD)	3.026	6.958	5.132	5.202	1.210
Proposed method	Max (Eucli,HD)	1.4e-05	3.2e-04	8.3e-05	6.1e-05	8.1e-05

Tab. 1 Results for 2D Damselfly wing shape example

- The optimum design of the 2D Stingray (contains 740 points in x and y coordinates) after 5,000 function evaluations.

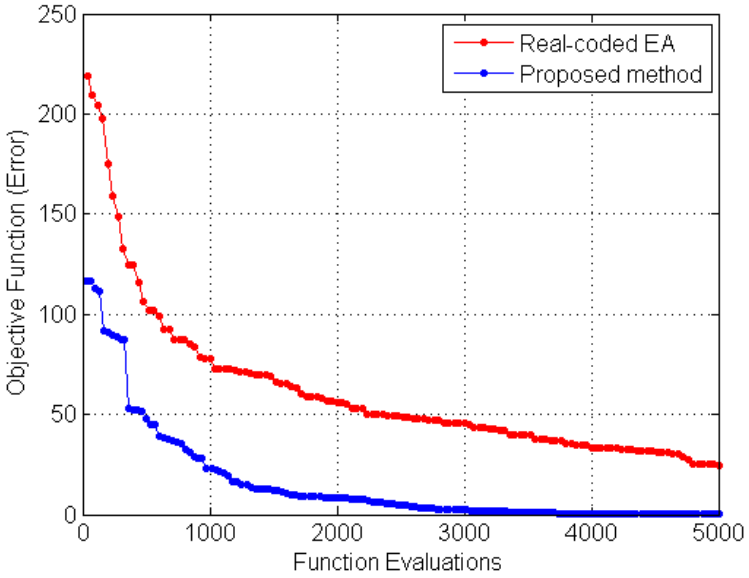
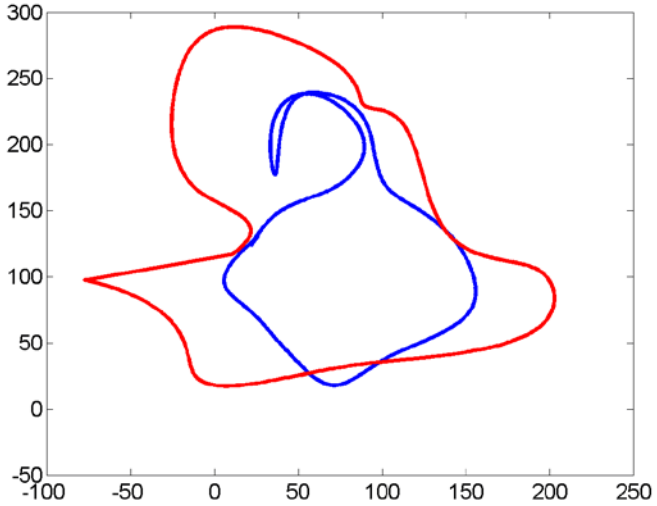


Fig.8 Progress plot of the best design for single objective matching error minimization

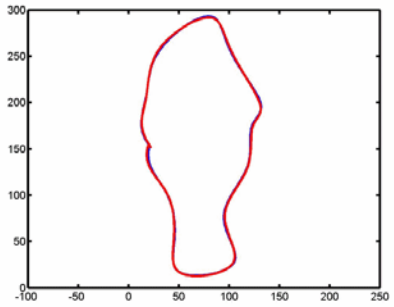


Ani.2 Evolution of Stingray (red) towards the target Stingray (blue)

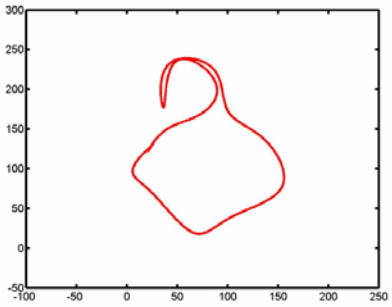
Method	Error Measurement	Best	Worst	Mean	Median	Std.
Real-coded EA	Max (Eucli,HD)	24.665	50.710	40.771	42.092	6.677
Proposed method	Max (Eucli,HD)	0.067	1.863	0.706	0.589	0.518

Tab. 2 Results for 2D Stingray shape example

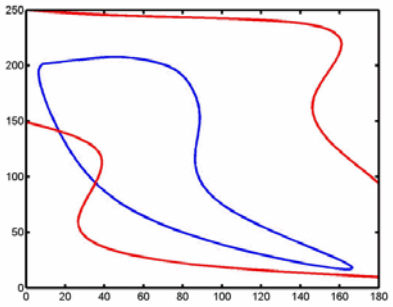
2D Examples



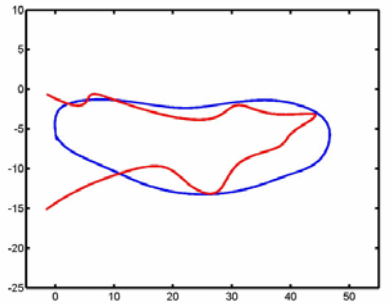
Simple fish



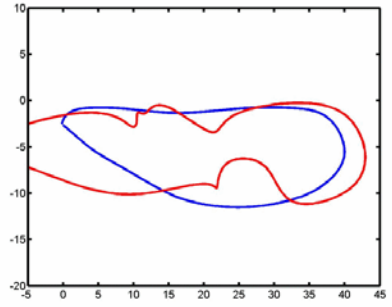
Stingray



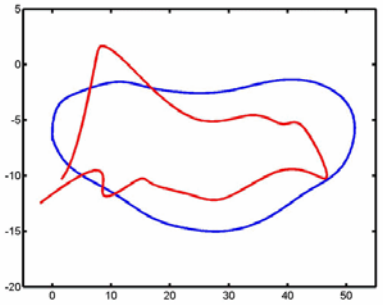
Airfoil



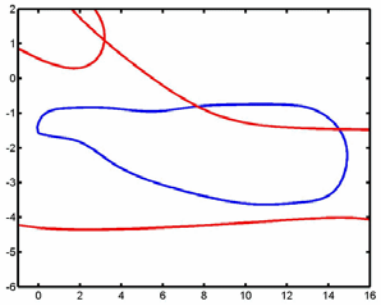
Dragonfly wing-1



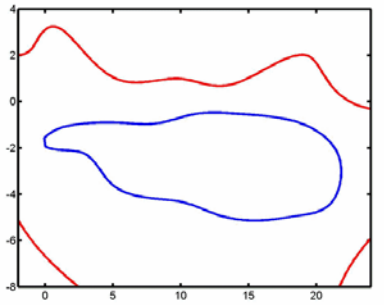
Dragonfly wing-2



Dragonfly wing-3



Damselfly wing-1



Damselfly wing-2

Fig.9 Evolution of generated shape (red) towards the target shape (blue)

3D Shape Representation and Optimization

Sorting Mechanism (3D)

- 3D shape is assumed to be disintegrated to 2D shape optimization problems termed as *Stations*.
- In order to avoid entangling of control points between two neighbouring stations (Fig.10), every set of control points spanning the shape in x direction is sorted in ascending order (Fig.11).

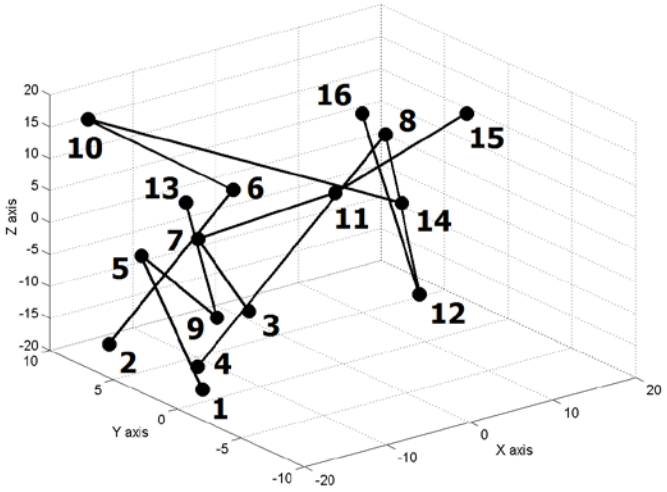


Fig.10 Position of control points before sorting

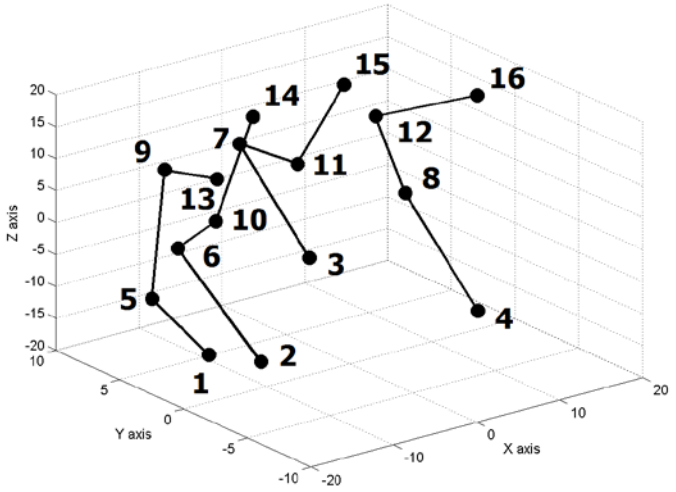


Fig.11 Position of control points after sorting

Khan, M., Mohamad, A.F.A., and Ray, T., An efficient memetic algorithm for 3D shape matching problems, *Engineering Optimization*, Under Review.

Performance of Smart Repair Strategies 3D

- The optimum design of the 3D Roger's surface (contains 289 points in x , y and z coordinates) after 10,000 function evaluations.

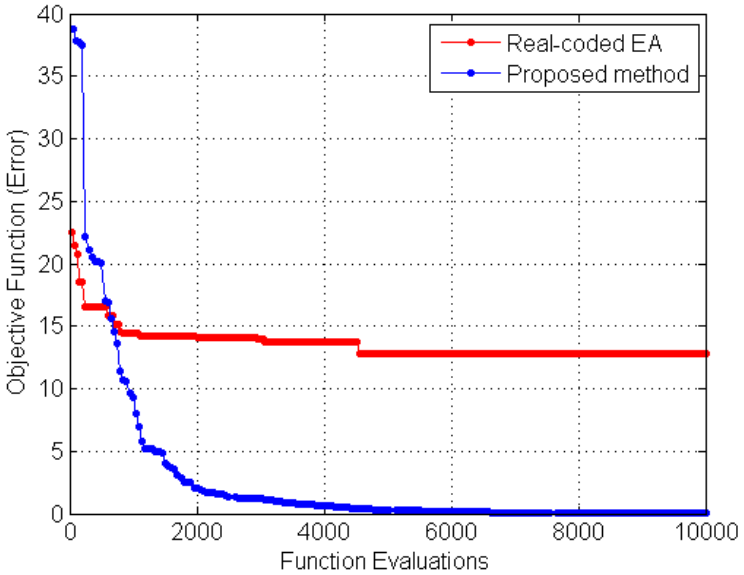
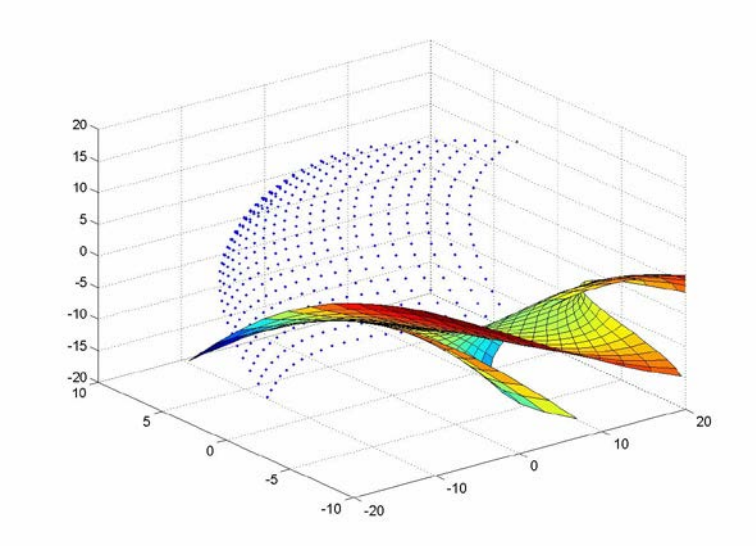


Fig.12 Progress plot of the best design for single objective matching error minimization



Ani.3 Evolution of 3D Roger's surface (evolving surface) towards the target Roger's surface (point cloud)

Method	Error Measurement	Best	Worst	Mean	Median	Std.
Real-coded EA	Max (Eucli,HD)	12.798	14.444	13.752	13.809	0.379
Proposed method	Max (Eucli,HD)	0.010	0.241	0.108	0.091	0.064

Tab. 3 Results for 3D Roger's surface shape example

Performance of Smart Repair Strategies 3D

- The optimum design of the 3D Flower vase (contains 676 points in x, y and z coordinates) after 100,000 function evaluations.

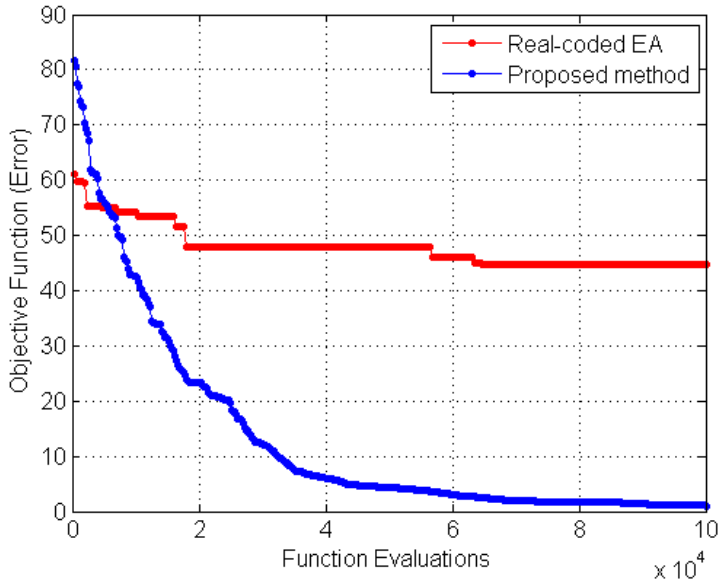
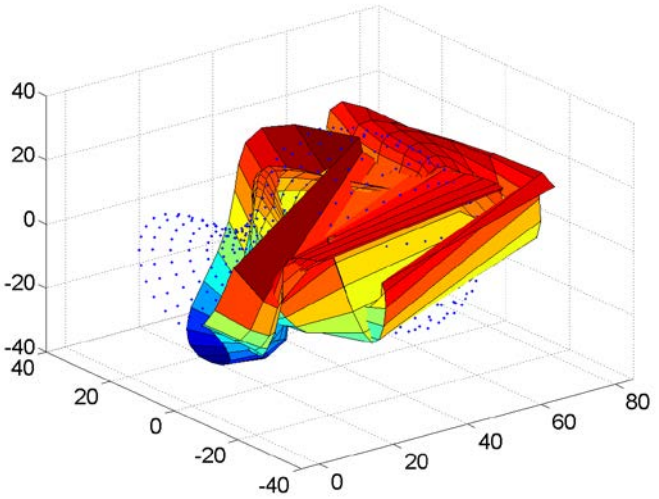


Fig.13 Progress plot of the best design for single objective matching error minimization



Ani.4 Evolution of 3D Flower vase (evolving surface) towards the target Flower vase (point cloud)

Method	Error Measurement	Best	Worst	Mean	Median	Std.
Real-coded EA	Max (Eucli,HD)	44.694	51.002	48.324	48.043	1.627
Proposed method	Max (Eucli,HD)	0.848	1.553	1.244	1.264	0.166

Tab. 4 Results for 3D Flower vase shape example

- The optimum design of the 3D Bulbous Bow (contains 441 points in x , y and z coordinates) after 50,000 function evaluations.

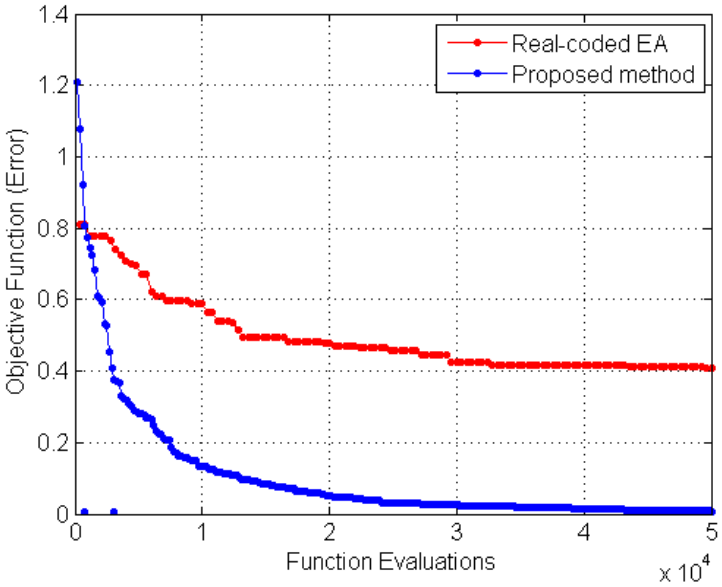
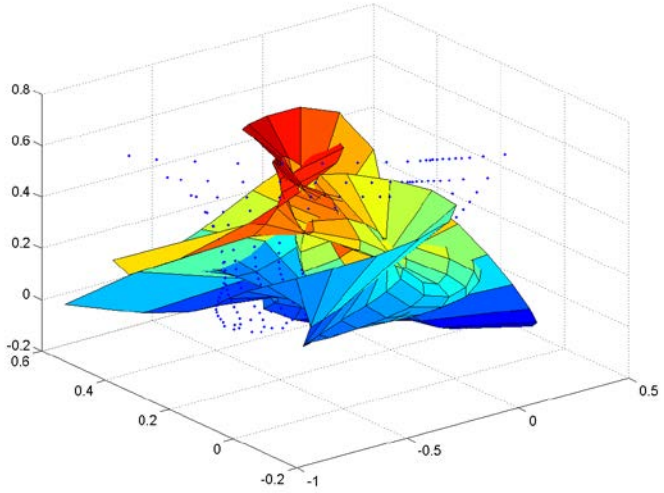


Fig.13 Progress plot of the best design for single objective matching error minimization



Ani.4 Evolution of 3D Bulbous Bow (evolving surface) towards the target Roger's surface (point cloud)

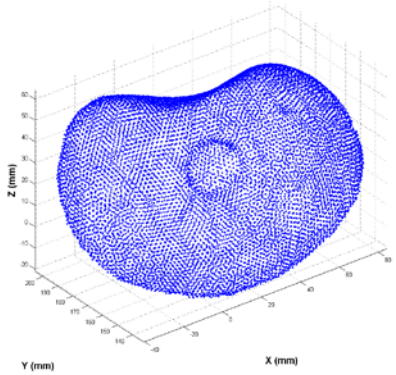
Method	Error Measurement	Best	Worst	Mean	Median	Std.
Real-coded EA	Max (Eucli,HD)	0.367	0.388	0.380	0.380	0.005
Proposed method	Max (Eucli,HD)	0.002	0.037	0.005	0.004	0.007

Tab. 4 Results for 3D Bulbous Bow shape example

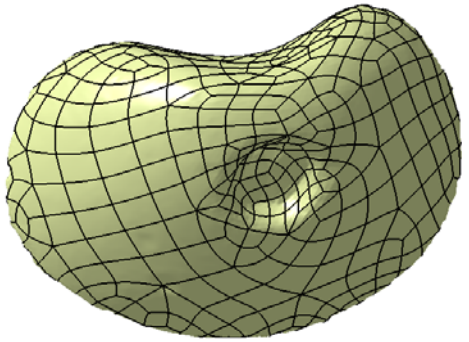
Real World Application

Shape Matching: 3D Real Objects

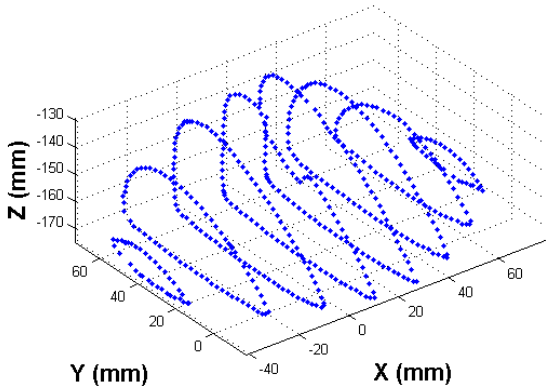
Rock Climbing Hold



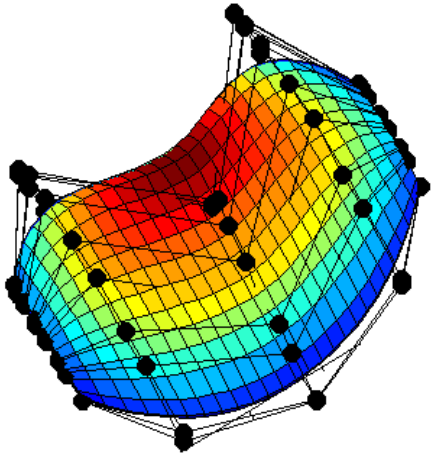
Data Cloud



IGES Surface



Post-processed Data Points



Inverse Fitting

Real World Application (3D Ear Plug)

- A 3D ear plug has been considered to support real world product application.
- The evolution of shapes is very crucial component for optimization of shapes according to a patient's ear anatomy and canal.
- The target shape is extracted in the form of a point cloud via a 3D scan.
- The optimum design of the 3D ear plug (contains 841 points in x, y and z coordinates) after 150,000 function evaluations.

Fig. 14 Ear plugs

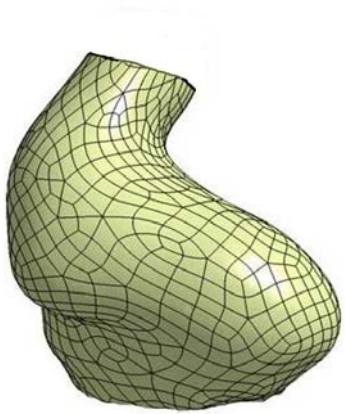


Fig. 15 3D scanned image

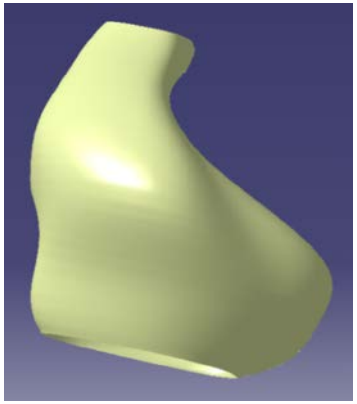
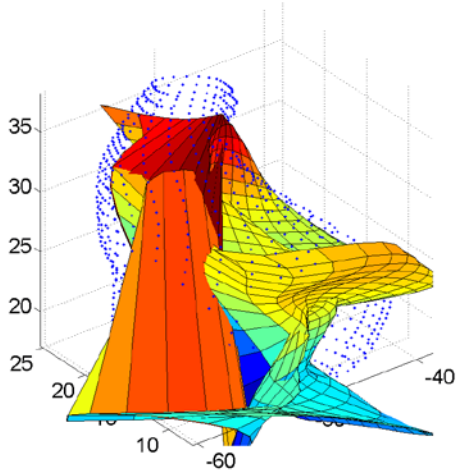


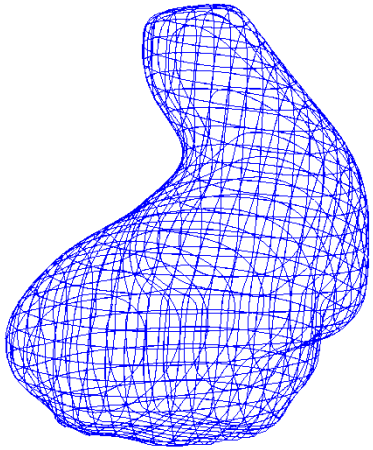
Fig.16 Target shape (CATIA model)



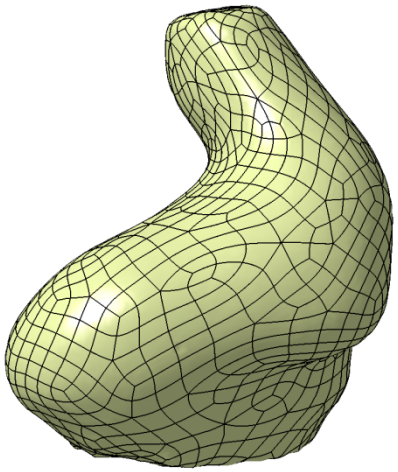
Ani.5 Ear plug evolutions

Shape Matching: Customized Ear Plugs

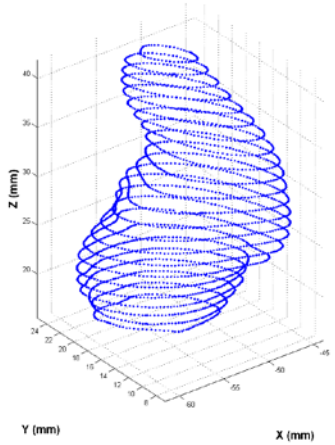
Hearing Aid



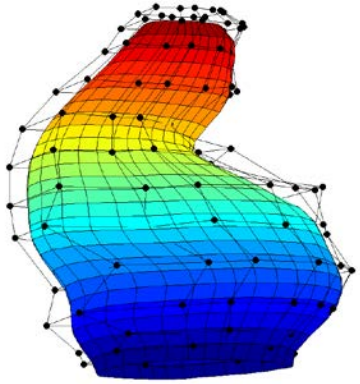
Data Cloud



IGES Surface



Post-processed Data Points



Inverse Fitting

Discussion and Conclusion

- An efficient method is introduced that is capable of solving 2D and 3D shape matching problems.
- The control points of the underlying B-spline representation are optimized using a memetic algorithm, while the presence of repair operation ensures generation of valid shapes.
- The memetic algorithm is embedded with multiple recombination strategies (EA and DE) and a local search to improve its efficiency.
- The proposed method has been tested using number of 2D shape examples of varying complexity, each of which was solved to desired accuracy within 5,000 evaluations, whereas for the complex 3D shape example 100,000 evaluations were required.
- The performance of the repair strategy has been studied for both 2D and 3D examples to highlight the benefits.
- Such a capability is the first step towards a cutting edge shape optimization approach, where one is interested to uncover novel shapes with extreme performance characteristics.

Challenges and Outlook

- ✓ Better algorithms to solve optimization problems with equality constraints.
- ✓ Test of Time: Many objective optimization problems and methods to solve them.
- ✓ Multi-fidelity Optimization
- ✓ Large Scale Optimization, i.e problems involving several hundreds of variables.
- ✓ Implementations on clouds and clusters.
- ✓ Knowledge embedded problem specific search strategies.
- ✓ Ensembles/Porfolio of methods with self adaptation.
- ✓ Learning to uncover design rules.

Application Snapshot: Underwater Vehicle Design

Introduction

- Unmanned Underwater Vehicles (UUVs):
  - Robotic mobile instrument carriers
  - Self-contained propulsion, sensors and intelligence
  - Complete sampling and survey tasks with little or no human intervention

- Application:
  - Scientific
    - Oceanographic survey
    - Search, classify, map
  - Commercial
    - Mineral field survey
    - Pipeline route survey
  - Military
    - Mine countermeasures
    - Anti-submarine warfare

- Types:
  - Autonomous Underwater Vehicle (AUV)
  - Remotely Operated Vehicle (ROV)
  - Autonomous Surface Vehicle (ASV)
  - Unpropelled underwater vehicle (Glider)

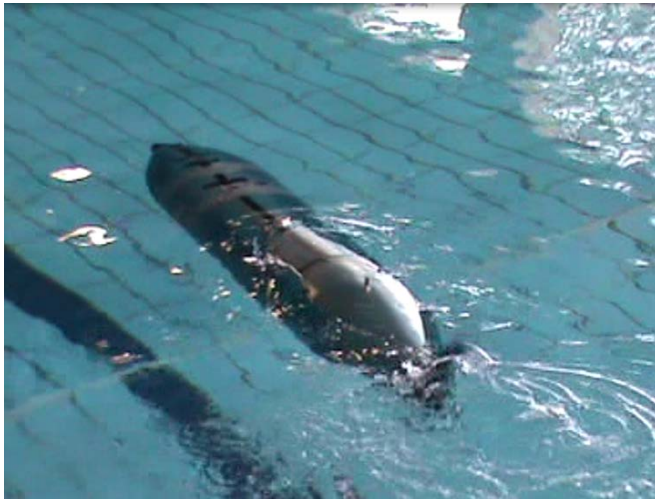


Fig. An autonomous underwater vehicle

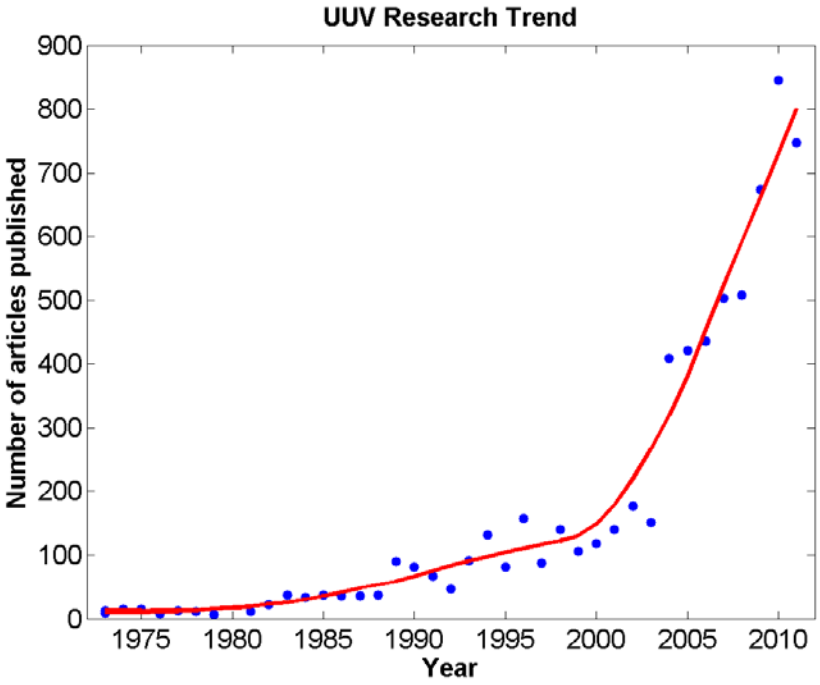


Fig. UUV research trend

*Source: ***Scopus***, database of academic journal articles*

Classification of UUVs

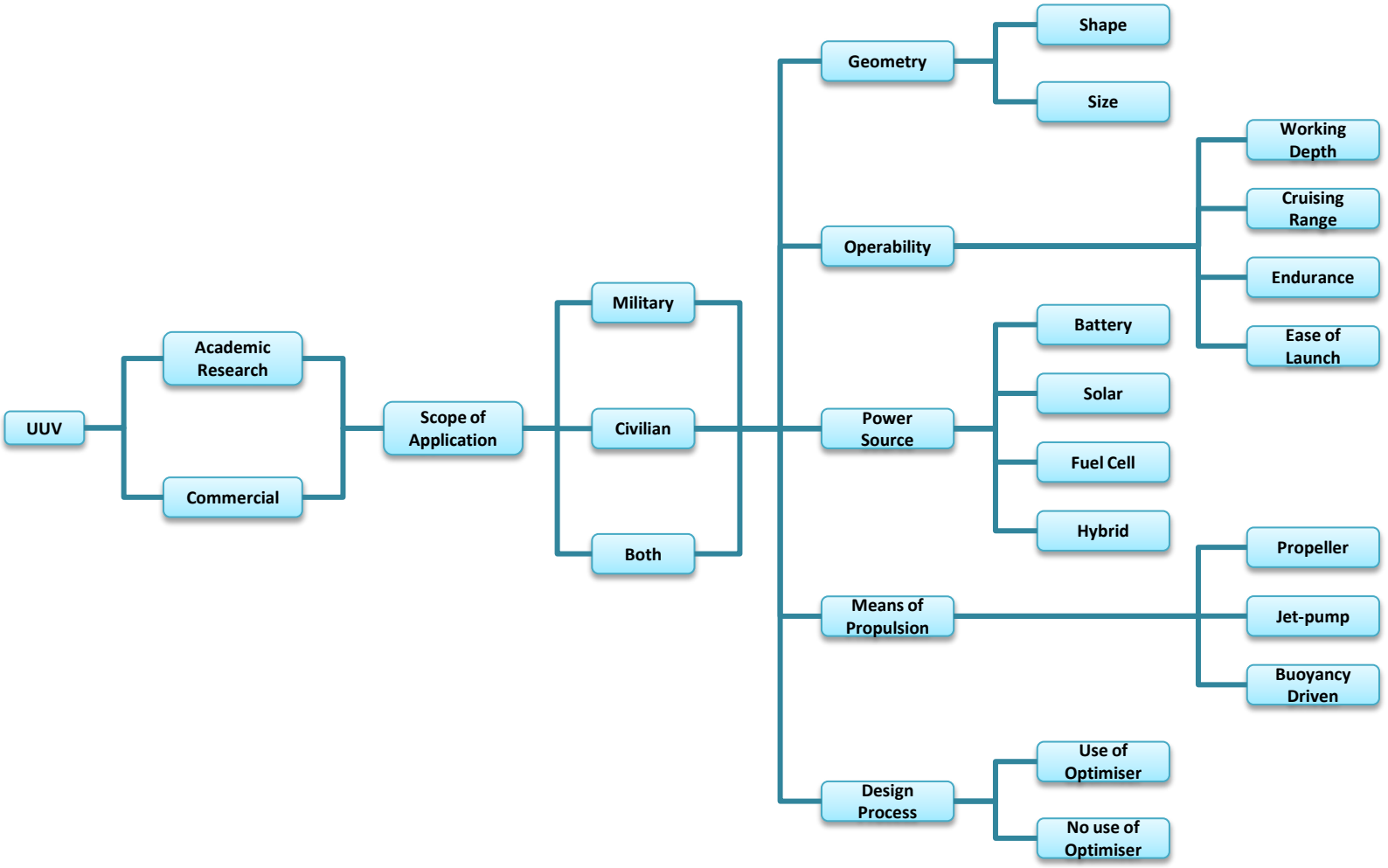
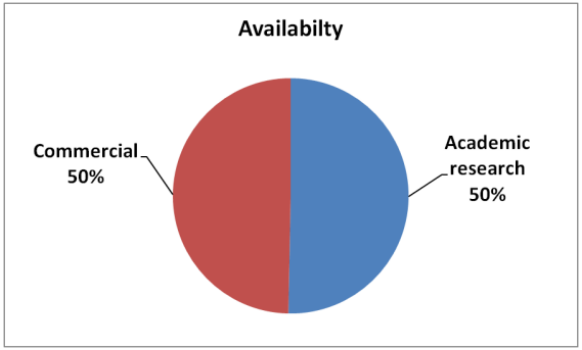
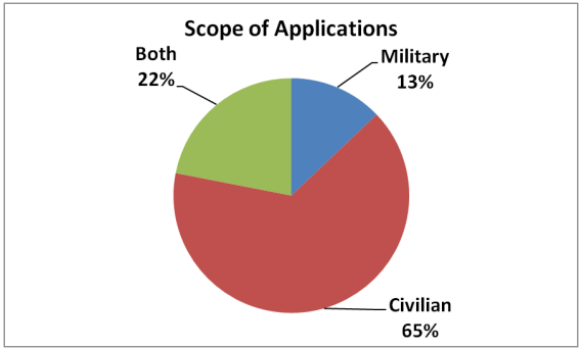
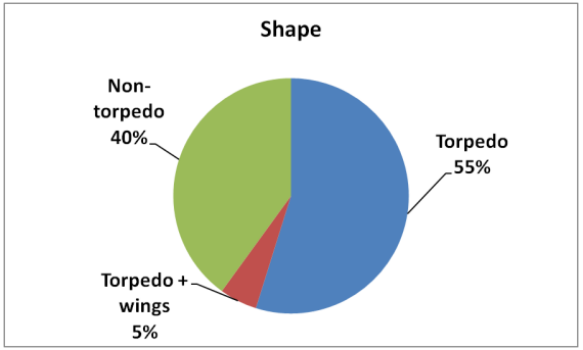


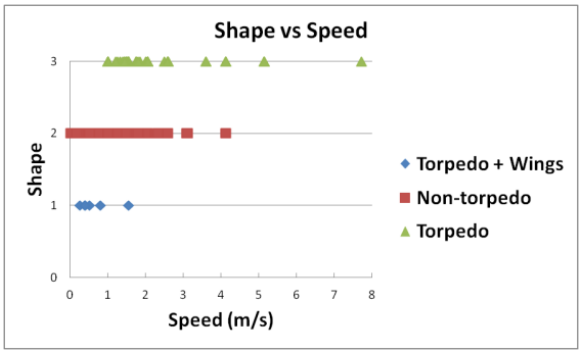
Fig. Classification tree of UUVs

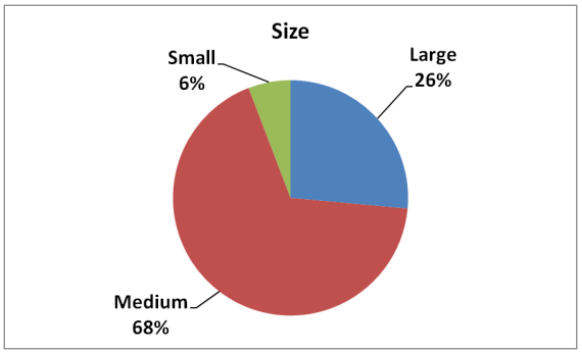
Classification of UUVs











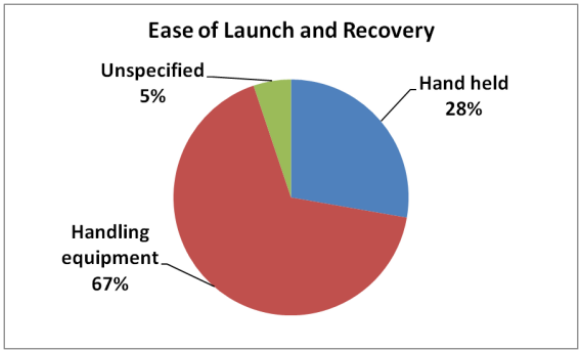
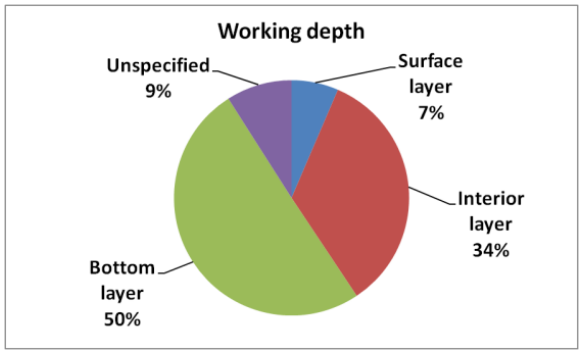
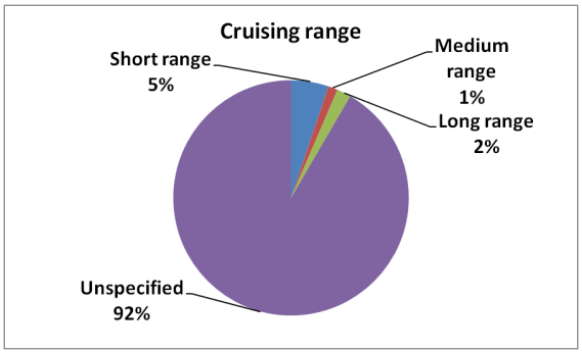
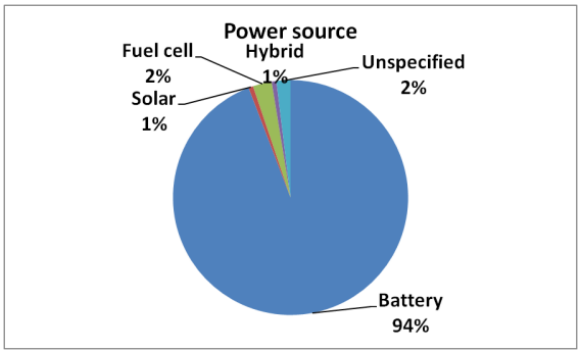


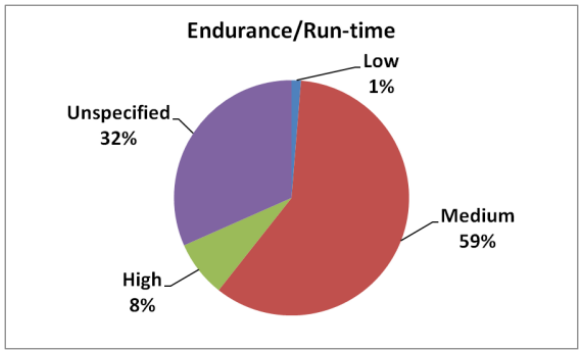
Fig. Classification of different UUVs under development and in use around the world

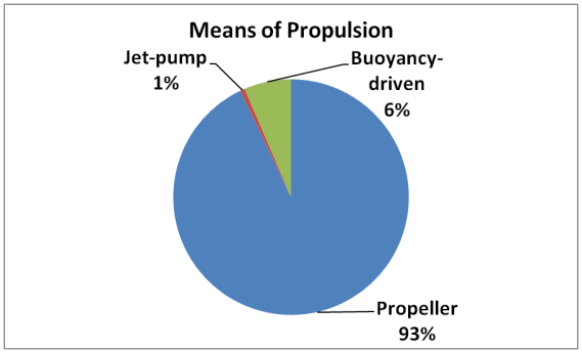
Classification of UUVs











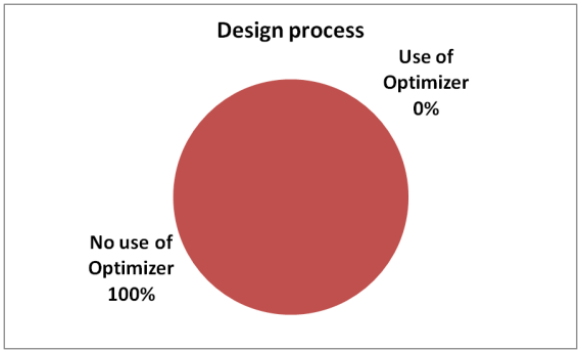


Fig. Classification of different UUVs under development and in use around the world (continued)

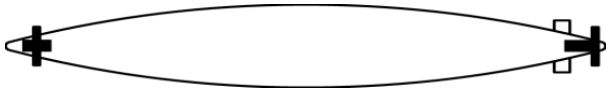
Motivation

- Very limited systematic study has been done with regards to optimum design.
- Previous attempts on UUV designs have focused primarily on functional designs, often accepting non-optimal designs.
- Placement strategy for clash-free arrangement of the on-board components has been overlooked by the researchers.
- Robust design, a design to mitigate the variation in design variables on the performance of the vehicle, has not been considered in the context of UUV designs.
- A model capable of handling different user and mission requirements.

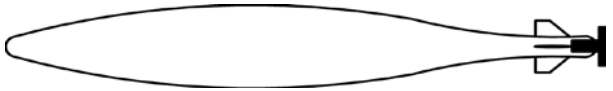
Approach

(a) Shape for good payload capacity

(b) Shape for low drag



(c) Shape for good manoeuvrability

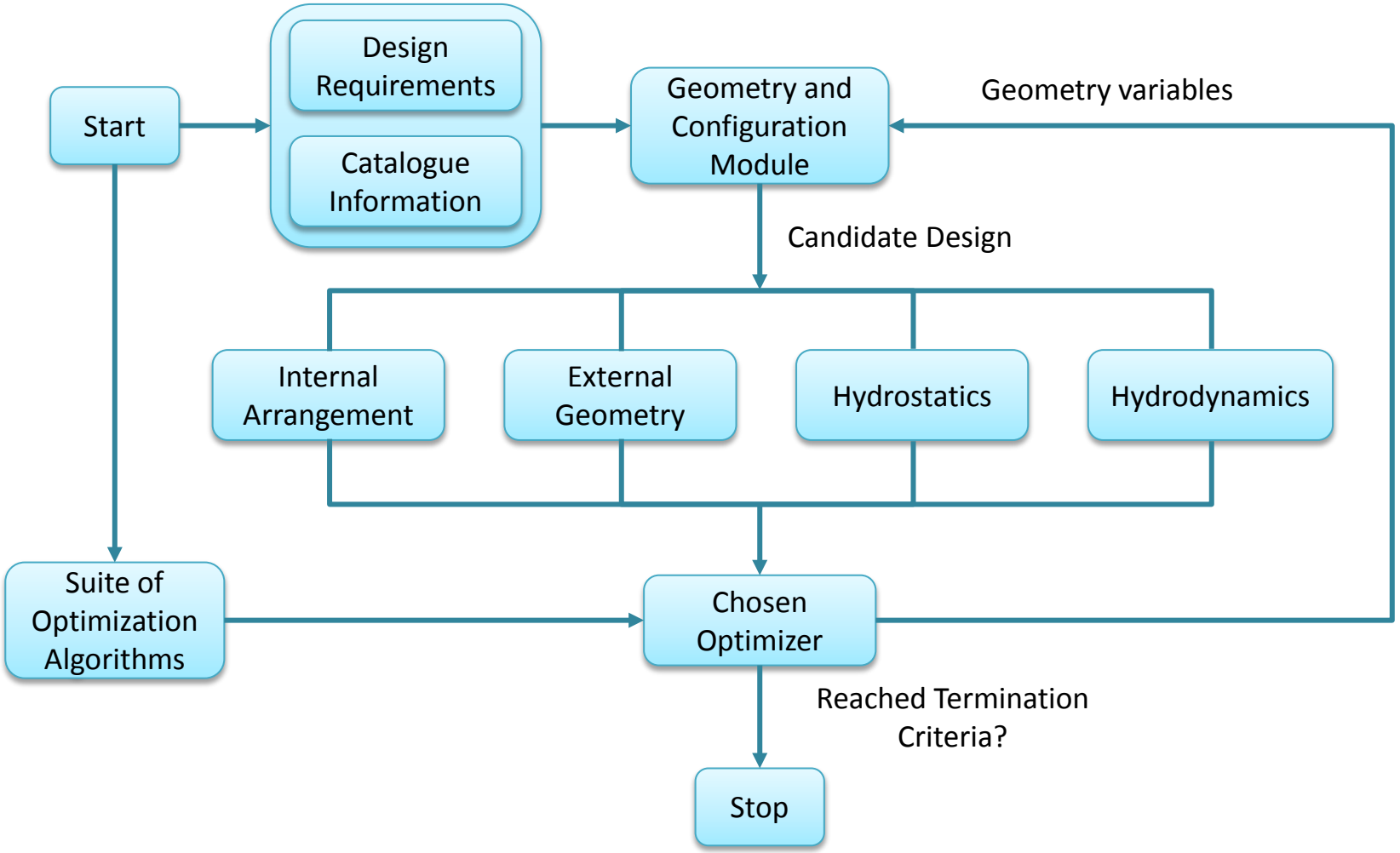


(d) Shape for high speed capacity

(e) Shape for good attitude control

Fig. Different shapes of UUV

Flowchart of the Optimization framework



Interfacing Multiple Analysis Tools

- The design optimization framework consists of several applications, namely Matlab, Microsoft Excel, CATIA, ANSYS (ICEM CFD) and FLUENT.
- **Matlab**: for numerical computation and is the basis of the whole optimization process.
- **Microsoft Excel** and **Text Document**: medium of communication between applications.
- **CATIA**: for modelling the vehicle based on the geometry parameters generated from the optimization process.
- **VBScript**: to automate CATIA modelling that can generate the model without user intervention.
- **ANSYS**: to simulate the flow around the designed vehicle.
  - **ICEM CFD**: to generate the mesh
  - **FLUENT**: to solve the flow problem
- The framework facilitates the communication of data from one application to the next through seamless integration of Matlab, CATIA and ANSYS; thereby producing an automated multidisciplinary design environment.

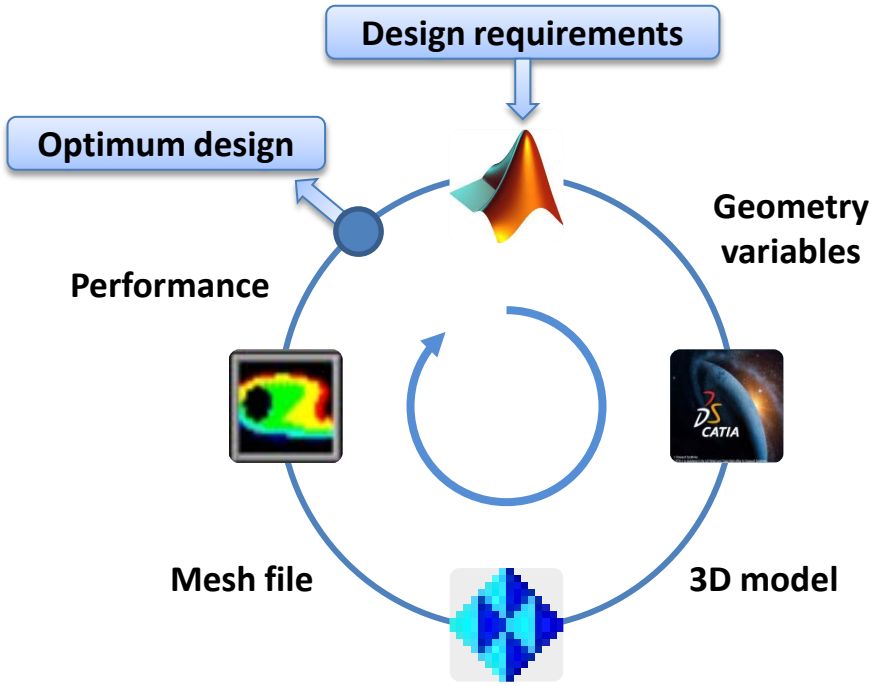


Fig. Inter-process communication flow among applications

Arrangement of Internal Components: Clash-free Mechanism

- An important aspect of this work is to apply a clash-free mechanism while arranging the internal parts of the UUV for optimal CG position.
- The term *clash-free* refers to the placement of the internal components such as controller, propellers and battery compartment in their respective positions in such a way that none of the components coincide (occupy the same place) either partially or wholly with other components, while maintaining appropriate clearance between them.
- As the design optimization approach is an iterative process, the use of the clash-free mechanism is essential to obtain a clash-free arrangement of the internal components for every valid design.
- Although in reality some components have irregular shapes, minimum bounding box dimensions have been used to derive the clash-free configuration.

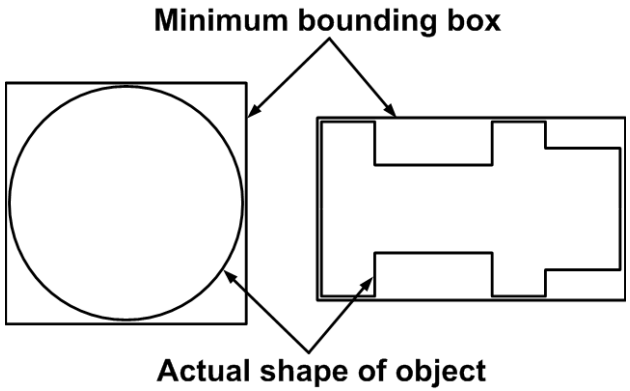


Fig. Illustration of bounding box concept used for internal components

Clash-free Mechanism

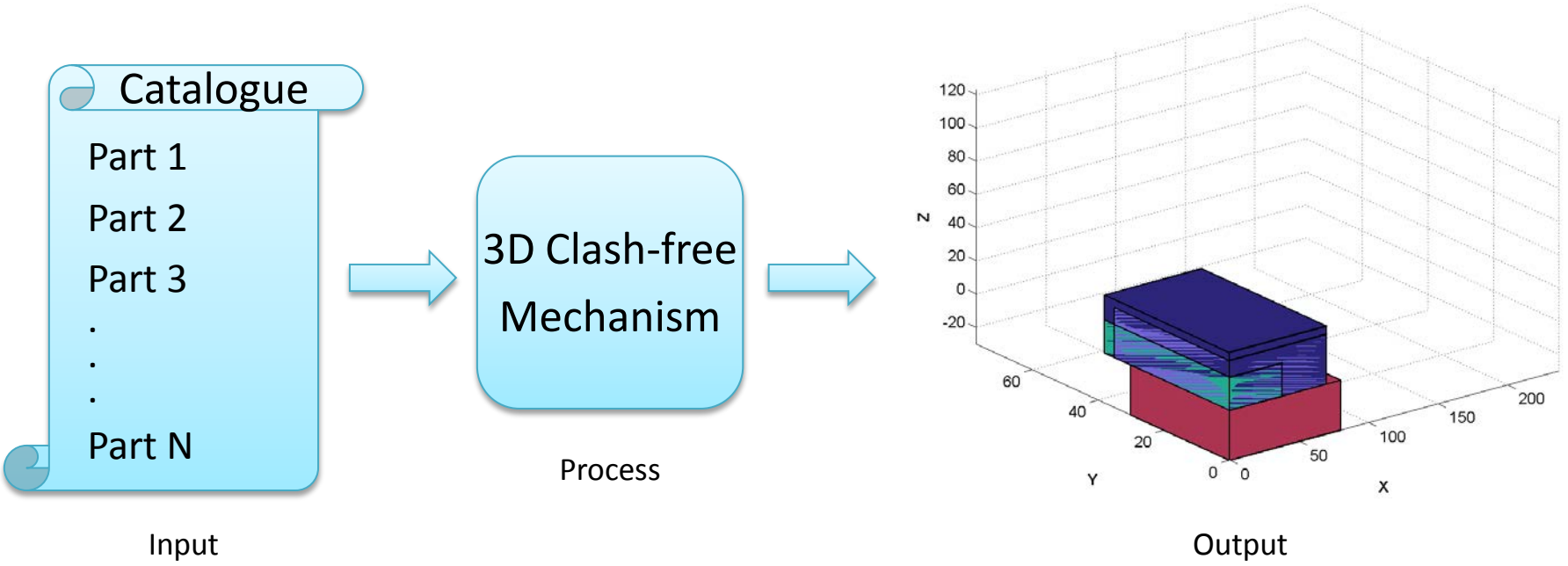


Fig. 3D clash-free mechanism

Algorithm: Clash-free Mechanism

Require: $O_1 : O_N$ (N >1) {Total number of internal parts to be placed}

Require: L_i (i = 1 : N) {Initial lower leftmost corner location of each part defined by the optimizer (x, y, z)}

Require: B_i (i = 1 : N) {Minimum bounding box dimensions of the parts (length, width and height)}

1: **for** i = 1 : (N − 1) **do**
2: Find coordinate matrix of O_i w.r.t. the global coordinate system, $C_i(8 \times 3) = L_i(8 \times 3) + B_i(8 \times 3)$
3: **for** j = (i + 1) : N **do**
4: Find coordinate matrix of O_j w.r.t. the global coordinate system, $C_j(8 \times 3) = L_j(8 \times 3) + B_j(8 \times 3)$
5: Create coordinate matrix of the pair of O_i and O_j w.r.t. the global coordinate system, $C_{ij}(16 \times 3) = [C_i(8 \times 3); C_j(8 \times 3)]$
6: Assign $D_i(1 \times 3) = \max(C_i) - \min(C_i)$, $D_j(1 \times 3) = \max(C_j) - \min(C_j)$
7: Compute bounding box dimensions of the pair, $D_{ij}(1 \times 3) = \max(C_{ij}) - \min(C_{ij})$
8: Compute clash, $M(1 \times 3) = D_i + D_j - D_{ij}$
9: Find how many elements of are non-zero, E = find(M >0) {E < 3 indicates no clash}
10: **if** (E = 3) **then**
11: Select the direction of movement (random or predefined in either X, Y or Z axis)
12: Move O_j, $C_j = C_j + M$, $L_j = L_j + M$ {New coordinates and location of O_j}
13: **while** (E = 3) **do**
14: Repeat steps 5 to 12, end if
15: **end while**
16: **end if**
17: **end for**
18: **end for**

Clash-free Mechanism

- An example of resolving a clash along the Z-direction (1D) using dimension comparison as adopted in the present approach is illustrated in the following figure.
- As can be seen, there are clashes between objects 1 and 2, and objects 2 and 3.
- Considering the first pair, the amount of clash between objects 1 and 2 is $c_{12}=l_1+l_2-l_{12}$. In figure, object 2 is moved along the Z-direction an amount equal to c_{12} and then the clash between objects 1 and 2 has been resolved.
- Similarly, the amount of clash between objects 2 and 3 can be found as $c_{23}=l_2+l_3-l_{23}$. Then object 3 is moved along the Z-direction by an amount equal to c_{23} and a clash-free arrangement is obtained as shown in the following figure.
- It is worth highlighting that in the developed algorithm there are provisions for selecting the object and the direction along which the selected object has to be moved.

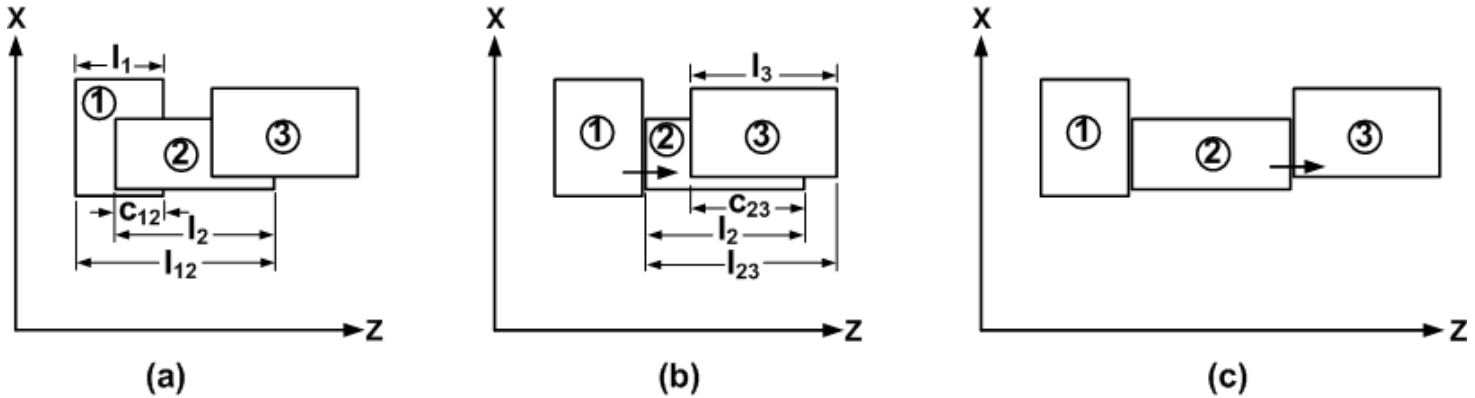


Fig. Typical illustration of the clash-free mechanism

- Two numerical approaches have been employed to estimate the drag of the designed vehicle:
  - Empirical estimation
  - CFD analysis

- **Empirical method**:

  The bare hull skin friction drag coefficient (C_F) as a function of Reynolds number (R_n), is found using the 1957 ITTC (International Towing Tank Conference) correlation line as,

$$C_F = \frac{0.075}{(\log_{10} R_n - 2)^2}$$

  where R_n is the Reynolds number and can be found as,

$$R_n = \frac{\rho V l}{\mu}$$

  where ρ is the density of the fluid, V is the velocity, l is the overall length and μ is the dynamic viscosity.

  Three methods: Virginia Tech (VT), MIT and G&J methods are employed in this study to compute the coefficient of viscous resistance (C_V) in three different ways, and the maximum drag value is chosen for further study.

According to G&J method, the coefficient of the viscous resistance, C_V , for the smooth bare hull can be found as,

$$C_V = C_F\left[1+0.5\left(\frac{d}{l}\right)+3\left(\frac{d}{l}\right)^3\right]$$

where d is the maximum body diameter

The VT method takes into account the effects of nose and tail shape variation coefficients, n_n and n_t respectively as,

$$C_V = C_F\left[1+0.5\left(\frac{d}{l}\right)+3\left(\frac{d}{l}\right)^{7-n_n-\frac{n_t}{2}}\right]$$

In MIT method, the prismatic coefficient, C_P , has been used to find the C_V as,

$$C_P = \frac{V_{env}}{\pi\left(\frac{d}{2}\right)^2 l}$$

where V_{env} is the displacement volume of the vehicle. Then,

$$C_V = C_F\left[1+1.5\left(\frac{d}{l}\right)^{\frac{3}{2}}+7\left(\frac{d}{l}\right)^3+0.002\left(C_P-0.6\right)\right]$$

Then the vehicle drag, D can be calculated as,

$$D = \frac{1}{2}\rho V^2 C_V S$$

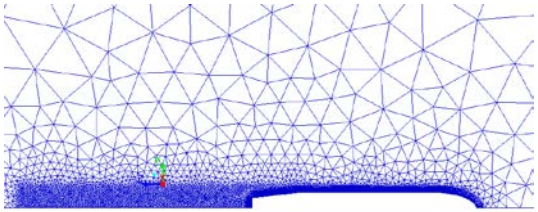
where S is the wetted surface area of the vehicle.

- **CFD analysis**:

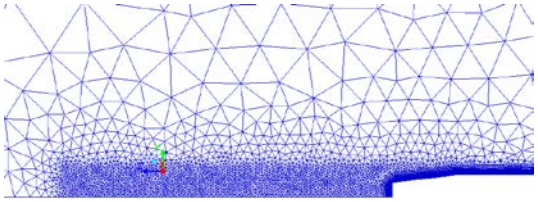
  The current work presents a 3D calculation of the total drag for the UUV. The adopted methodology and the pre-processing for the CFD analysis are:

  - *Model building and flow domain*:
    - The bare hull shape is produced in CATIA based on the hull parameters and exported to ICEM CFD as a *step* file for meshing.
    - In case of an axisymmetric UUV, quarter model of the bare hull is considered for CFD analysis in order to reduce the computational costs.
  - *Mesh generation*:
    - After defining the model and the far field, the solution domain is decomposed into appropriate number of locations based on the accuracy of the results required.
    - Mesh is generated using ICEM CFD Meshing software.
    - To automate the mesh generation process, scripting is done for the ICEM CFD written in Tcl/Tk language. The main steps for automated mesh generation process are:
      - Import the geometry (CATIA generated .stp) file
      - Repair the geometry (build topology)
      - Define the flow domain and the fluid material point
      - Define the surfaces of the flow domain and the geometry, such as inlet, outlet, side walls, symmetry planes, vehicle body, etc
      - Define the mesh parameters
      - Build mesh and check mesh
      - Export the mesh with boundary conditions as a '.msh' file

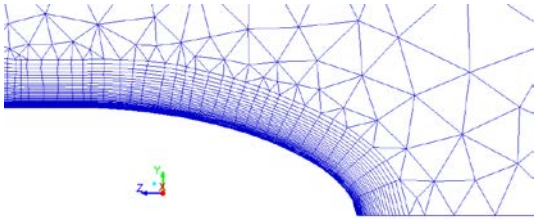
Analysis Module: Drag Estimation



(a) Mesh of the flow domain



(b) Dense mesh near the stern



(c) Boundary layer grid

Fig. Typical illustration of the CFD analysis

- *Model selection and simulation*:
  - When the computational domain is meshed, the flow is solved using the software ANSYS FLUENT.
  - To automate the problem setup and the solution process, another script file has been written to run the program in batch mode. The main steps for the solution process are:
    - Import the ICEM CFD generated mesh (.msh) file
    - Check the mesh
    - Select the model, e.g. k – ε model
    - Define the cell zone conditions and flow material, e.g. water-liquid
    - Define the boundary conditions
    - Define the solution methods and convergence criteria
    - Initialize the solution
    - Run calculation
    - Export the drag results

Optimization Module

- The design optimization framework is embedded with a suite of state-of-the-art optimization algorithms, namely:
  - Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb *et al.* 2002)
  - Infeasibility Driven Evolutionary Algorithm (IDEA) (Singh *et al.* 2008)
  - Infeasibility Empowered Memetic Algorithm (IEMA) (Singh *et al.* 2010)
- It is worth mentioning that any optimization algorithm capable of solving single and multi-objective optimization problems can be used within the framework.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, 6 (2), pp. 182–197, 2002.

Singh, H.K., Isaacs, A., Ray, T., and Smith, W., "Infeasibility Driven Evolutionary Algorithm (IDEA) for Engineering Design Optimization," 21st Australasian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence, Auckland, NewZealand (2008), Springer-Verlag Berlin Heidelberg, LNAI 5360, pp. 104–115, 2008.

Singh, H.K., Ray, T., and Smith, W., "Performance of Infeasibility Empowered Memetic Algorithm (IEMA) on Engineering Design Problems," Australasian Conference on Artificial Intelligence'2010, pp. 425–434, 2010.

Process of the Design Optimization

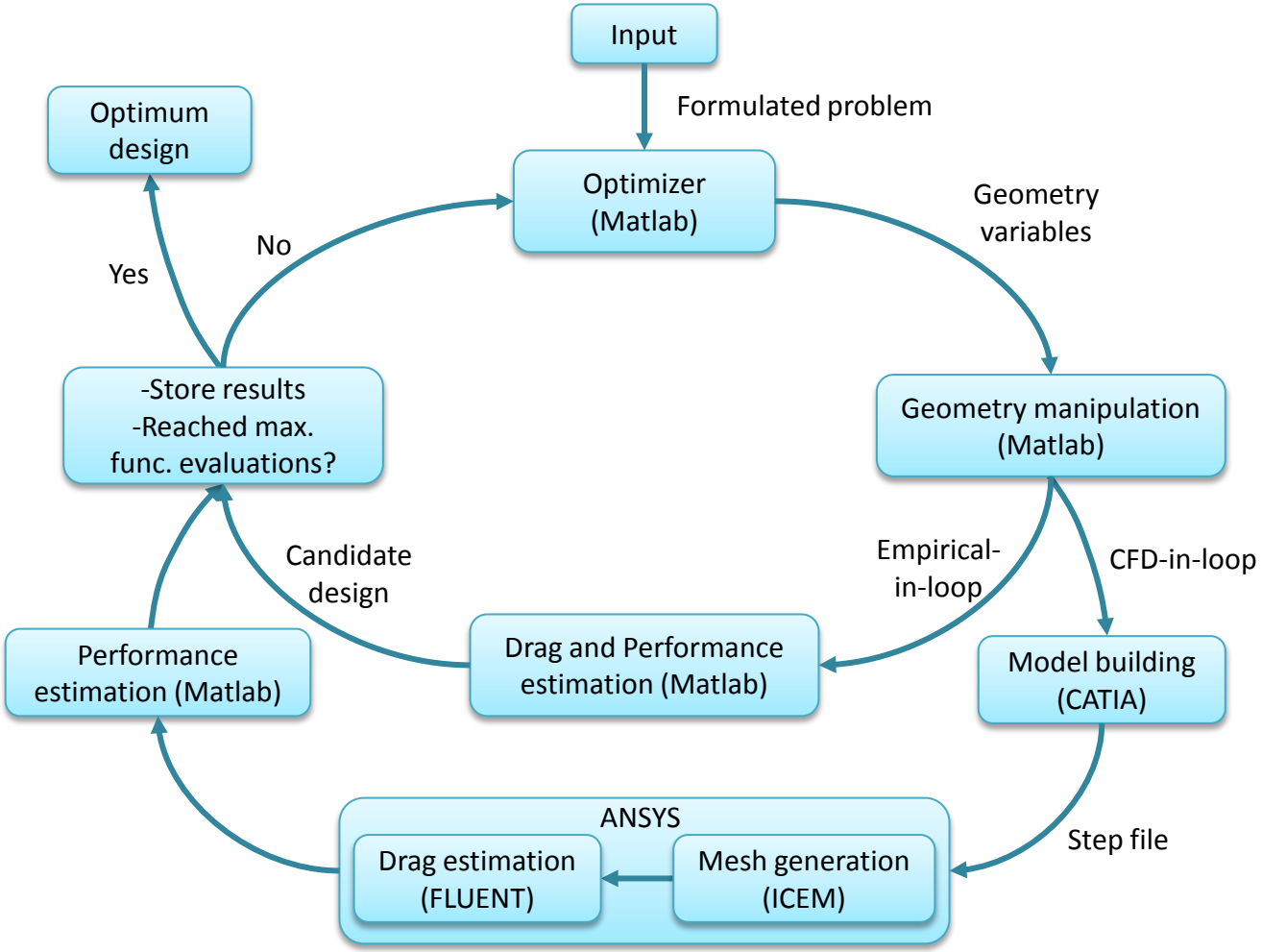


Fig. A diagram showing the interaction among modules and the flow of data from one application to the next for the design optimization framework

Process of the Design Optimization

- The design optimization process starts with a set of design variables that is fed into the optimization module.
- The optimizer generates candidate design with the values of design variables within the bound that are subsequently used to generate the hull form geometries.
- The geometry and configuration modules not only generate the external hull geometry but also place the internal on-board components in a clash-free state.
- Once the internal parts are placed in a clash free state, the parallel middle body is generated automatically covering the internal arrangement, and then nose, rear propeller and tail cone are attached along with the mid-body, thereby generating the complete vehicle shape.
- The performance of the candidate design is evaluated and then used by the optimizer.
- The process essentially involves two level of fidelity models: low fidelity model (empirical-in-loop) and high fidelity model (CFD-in-loop).
  - low fidelity simulations are less time consuming but could be less accurate.
  - high fidelity simulations are often more time consuming, however, more accurate.
- In the present study, for CFD-in-loop process, Matlab is interfaced with CATIA and the CFD software ANSYS through scripting.

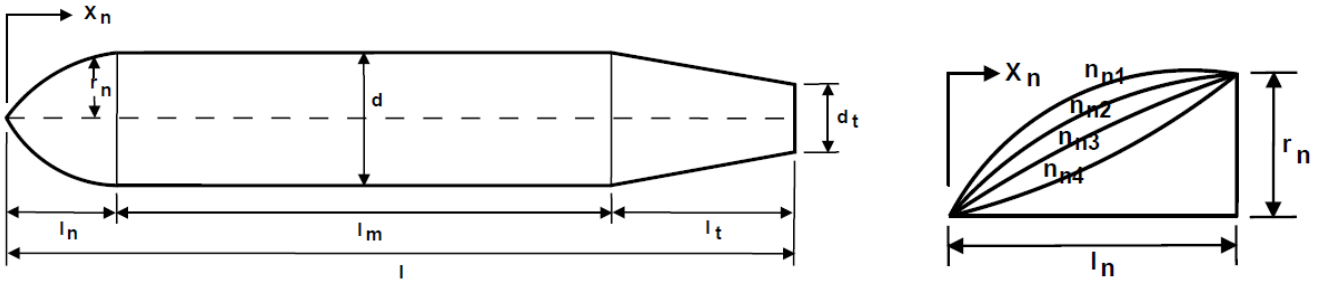
Design Optimization of a Model Submarine : Toy Sub

Approach 1: Optimization using Low and High fidelity Models

Approach 2: Robust Design Optimization

Toy Sub: Design Requirements

- Operating speed should be 0.5 m/s
- No longer than 500 mm
- Total weight be less than or equal to 500 g
- The vehicle to be propelled by one rear propeller of dimension $30 \times 30 \times 80$ mm and two propellers each for pitch and yaw movements, respectively, of dimension $70 \times 35 \times 30$ mm each. The individual weight of the rear propeller and the propellers for pitch and yaw movements is same, and the value is 30 g
- Should have enough free space to carry a controller and a battery unit of dimensions $38 \times 38 \times 20$ and $70 \times 38 \times 35$ mm, respectively. The weights of the controller and battery units are 45 and 120 g, respectively.



(a) Hull geometry (b) Nose geometry

Fig. Parameterization of the vehicle geometry

- Equation of the nose:

$$r_n = \frac{1}{2} d \left[1 - \left(\frac{l_n - x_n}{l_n} \right)^{n_n} \right]^{\frac{1}{n_n}}$$

where r_n is the radius of the nose, d is the maximum body diameter, which may be varied, l_n is the length of the nose, x_n is the reference length that varies from 0 to l_n, and n_n is the shape variation coefficient of the nose which may also be varied to give different shapes of the nose.

Toy Sub: Propulsion System

- The propulsion of the toy submarine is achieved through the use of three propeller units:
  - One is for vertical movement: up and down
  - One is for lateral movement: left and right
  - One is to propel the vehicle: forward and backward

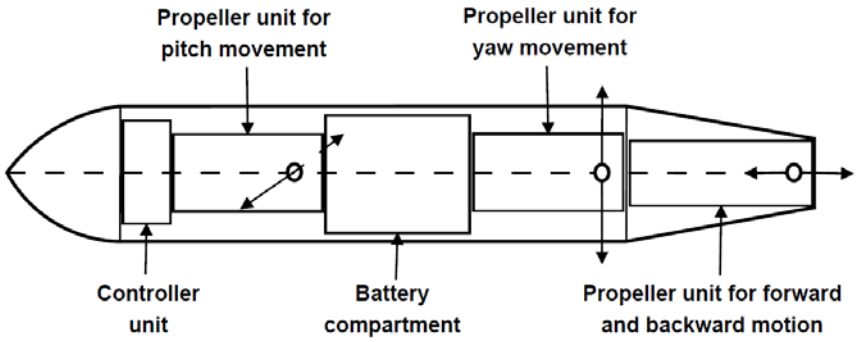


Fig. Configuration of the propulsion system

Approach 1: Optimization Using Low and High Fidelity Models

- **Single Objective Optimization Problem Definition**

Minimize :

$f(1) = D$

Subject to :

$g(1) = l \le 500\,mm;\ g(2) = w \le 500g$

$g(3) = LA_1 \ge 45\,mm;\ g(4) = LA_2 \ge 90\,mm$

$g(5) = s \le 4\,mm;$

Variable bounds :

$0 \le Z_C \le 300\,mm;\ 0 \le Z_V \le 300\,mm$

$0 \le Z_B \le 300\,mm;\ 0 \le Z_L \le 300\,mm$

$35 \le d_t \le 50\,mm;\ 80 \le l_t \le 150\,mm$

$1.5 \le n_n \le 3;\ 45 \le l_n \le 100\ mm$

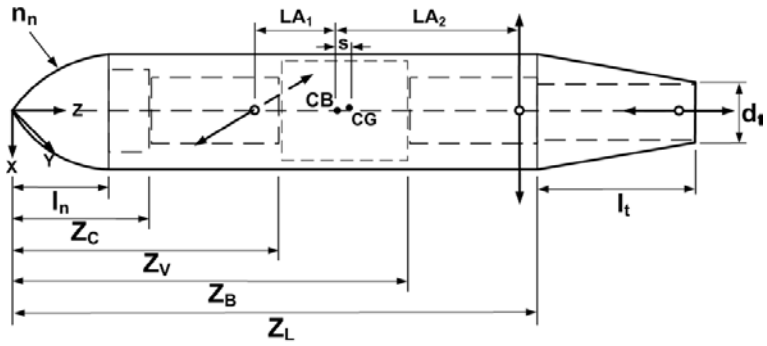


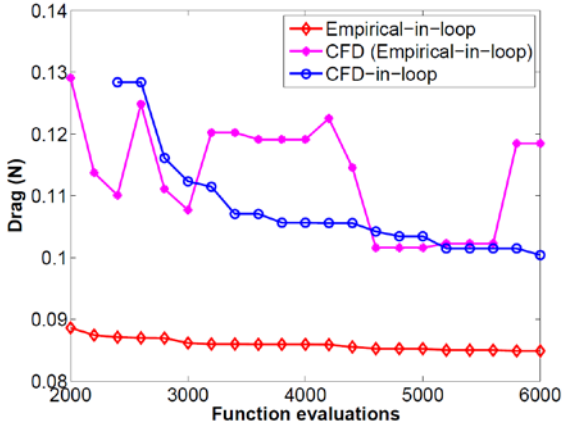
Fig. The constraints and design variables for problem formulation

- The numerical simulation can be run at two level of fidelity models dictated by the accuracy of drag estimation.
- The computing time per evaluation for low and high fidelity models is about 0.05 s and 347.54 s respectively, on a Intel Xeon processor machine of 3.33 GHz with 6.00 GB memory.

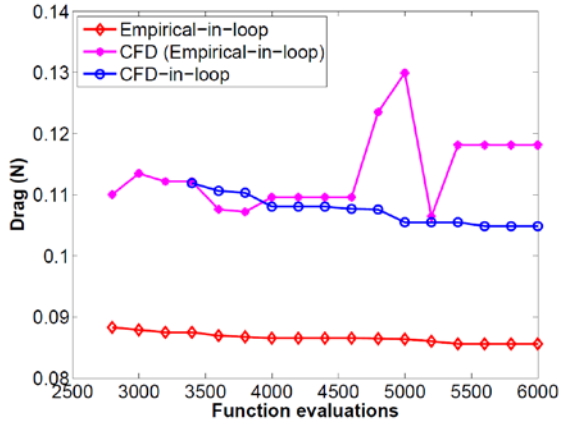
Approach 1: Results

Table: Single objective drag minimization results

Drag	Empirical-in-loop	CFD-in-loop
Best (N)	0.0848660	0.1003834
Mean (N)	0.0854686	0.1039560
Median (N)	0.0855750	0.1048310
Worst (N)	0.0858490	0.1057080
SD (N)	0.0004181	0.0021912



(a) Best design



(b) Median design

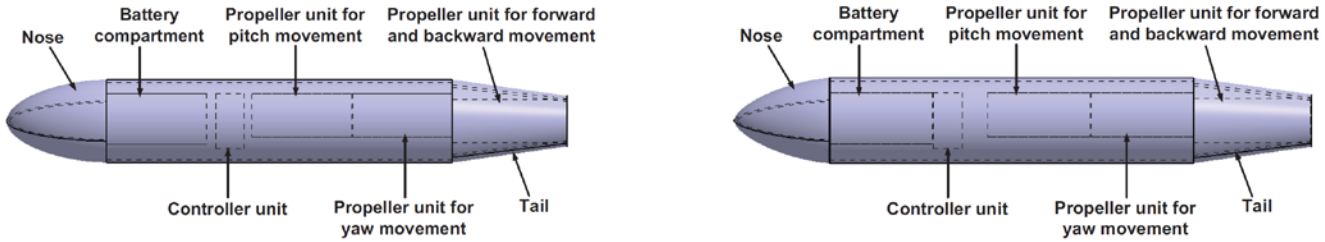
Fig. Progress plots of the best and median designs using IDEA

Approach 1: Results

- The best solutions of the two fidelity models are referred as the **Design-I** (best solution of the empirical-in-loop analysis) and **Design-II** (best solution of the CFD-in-loop analysis) toy submarines.

(a) Design-I (b) Design-II

Fig. CATIA models of the designed toy submarines



(a) Design-I (b) Design-II

Fig. Configurations of the designed toy submarines

(a) Design-I (b) Design-II

Fig. Longitudinal sections of the designed toy submarines

Approach 1: Results

- An example of a similar existing submarine available in the market is USS Dallas RC toy submarine.
- This model submarine has independent propellers to allow it to ascend, descend, turn and move forward and backward.
- The designs identified through the process of optimization are compared with the existing toy submarine to highlight the benefits offered by the present approach.

Fig. USS Dallas RC toy submarine

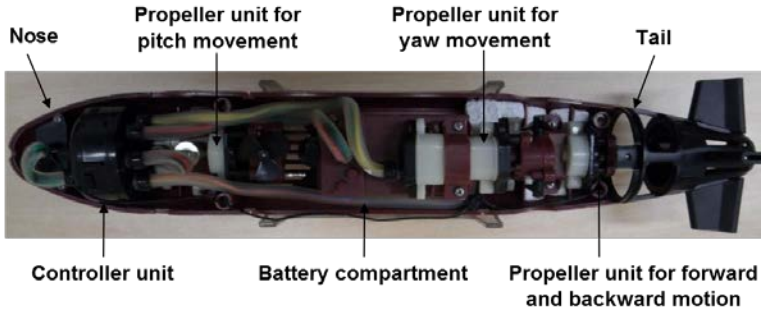


Fig. Configuration of the USS Dallas RC toy submarine

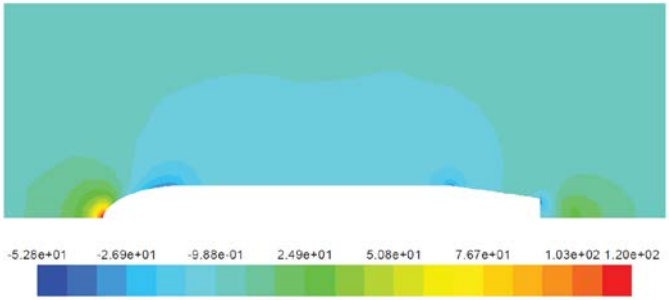
Approach 1: Results

Table: Performance criteria of the existing USS Dallas RC toy, Design-I and Design-II submarines

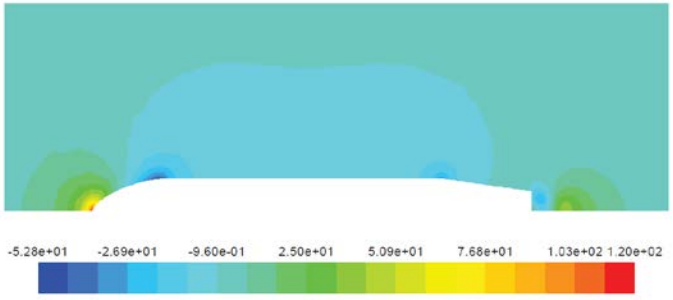
Vehicle particulars	USS Dallas	Design-I	Design-II
Nose length	45 mm	68 mm	63 mm
Mid-body length	210 mm	242 mm	248 mm
Tail length	95 mm	80 mm	80 mm
Length overall	350 mm	390 mm	391 mm
Outer diameter	60 mm	58 mm	58 mm
L/D ratio	5.8	6.7	6.7
Wetted surface area	0.082385 m^2	0.088521 m^2	0.088513 m^2
Displacement volume	0.000437 m^3	0.000493 m^3	0.000480 m^3
Displaced water mass	437 g	493 g	480 g
Total mass of the vehicle	430 g	442 g	441 g
First lever arm length	45 mm	45 mm	45 mm
Second lever arm length	90 mm	112 mm	112 mm
CG-CB separation	3.705 mm	3.729 mm	1.123 mm
Nominal speed	0.5 m/s	0.5 m/s	0.5 m/s
Drag (Empirical estimation)	0.0825860 N	0.0848660 N	0.0847710 N
Drag (CFD analysis)	0.1065059 N	0.1184621 N	0.1003834 N

Approach 1: Results of CFD Analyses

- CFD analysis provides not only an accurate simulation of the flow around the vehicle but also a useful understanding of the fluid-structure interactions.

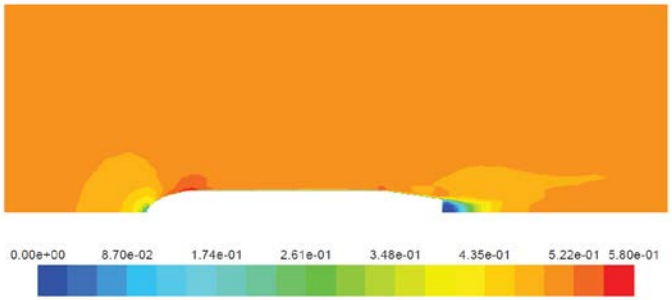


(a) Design-I

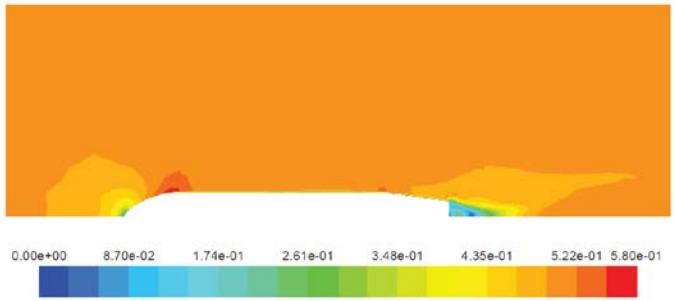


(b) Design-II

Fig. Pressure contours around the surfaces of the designed toy submarines



(a) Design-I



(b) Design-II

Fig. Velocity contours around the surfaces of the designed toy submarines

Approach 2: Robust Design Optimization

- An optimal design is often of less practical use if its performance is likely to degrade significantly under expected variations in variables or operating conditions.
- Furthermore, often such optimal designs lie on constraint boundaries which tend to fail or become unsafe under such expected variations.
- Therefore, for a practical design, there is a need to identify optimum solutions to the constrained robust optimization problem
- The framework is subsequently used to identify optimal and robust optimal designs of a small-scale (length nominally less than 500 mm) and light-weight (less than 0.5 kg) toy submarine.

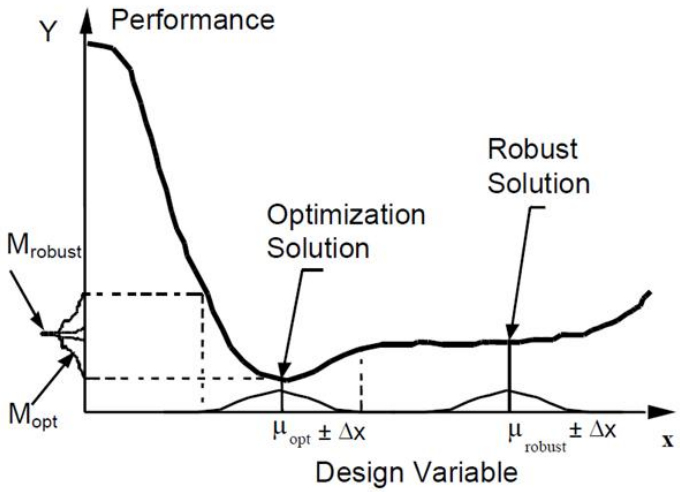


Fig. Illustration of Type II robust design optimization

Alam, K., Ray, T., and Anavatti, S., "A new robust design optimization approach for unmanned underwater vehicle design," *Proc IMechE Part M: Journal of Engineering for the Maritime Environment,* 226(3), pp. 235-249, 2012.

Approach 2: Problem Formulations

- **Single Objective Optimization Problem Definition**

$$\textit{Minimize}:$$
$$f(1) = D$$
$$\textit{Subject to}:$$
$$g(1) = l \le 500\,mm;\ g(2) = w \le 500g$$
$$g(3) = LA_1 \ge 45\,mm;\ g(4) = LA_2 \ge 90\,mm$$
$$g(5) = s \le 4\,mm;$$

$$\textit{Variable bounds}:$$
$$0 \le Z_C \le 300\,mm;\ 0 \le Z_V \le 300\,mm$$
$$0 \le Z_B \le 300\,mm;\ 0 \le Z_L \le 300\,mm$$
$$35 \le d_t \le 50\,mm;\ 80 \le l_t \le 150\,mm$$
$$1.5 \le n_n \le 3;\ 45 \le l_n \le 100\ mm$$

- **Robust Design Optimization Problem Definition**
  - To assess the robustness of a solution, a 2% uniform variation is considered for each variable and the mean objective value is computed using 100 solutions within the above neighbourhood.
  - The objective function is the minimization of mean value of the drag. In addition to the constraints listed above, three additional constraints, i.e. number of violations (*Ng*) in each of the constraints among the neighbouring solutions, are considered. The design variables are the same as for single objective optimization problem.

$$\textit{Minimize}:$$
$$f(1) = D_m$$
$$\textit{Subject to}:$$
$$g(1) = l \le 500\,mm;\ g(2) = w \le 500g$$
$$g(3) = LA_1 \ge 45\,mm;\ g(4) = LA_2 \ge 90\,mm;\ g(5) = s \le 4\,mm$$
$$g(6) = N_{g1} \le 0;\ g(7) = N_{g2} \le 0;\ g(8) = N_{g3} \le 0$$

Approach 2: Results

Table: Single objective drag minimization results

Design	Optimum	Robust
Best (N)	0.082177	0.0921119
Mean (N)	0.083828	0.0979447
Median (N)	0.084119	0.1000550
Worst (N)	0.086806	0.1207190
SD (N)	0.000892474	0.0071278

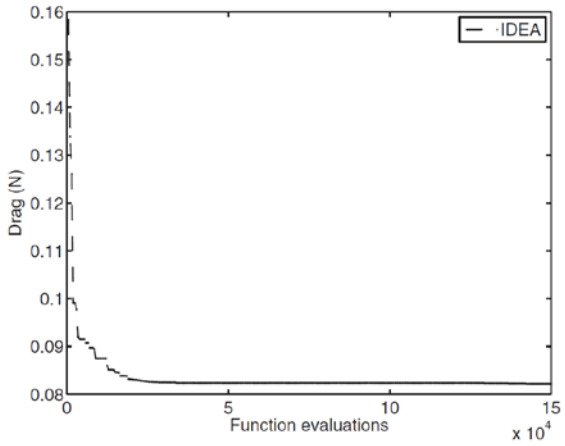


Fig. Progress plot of the best design of optimization approach

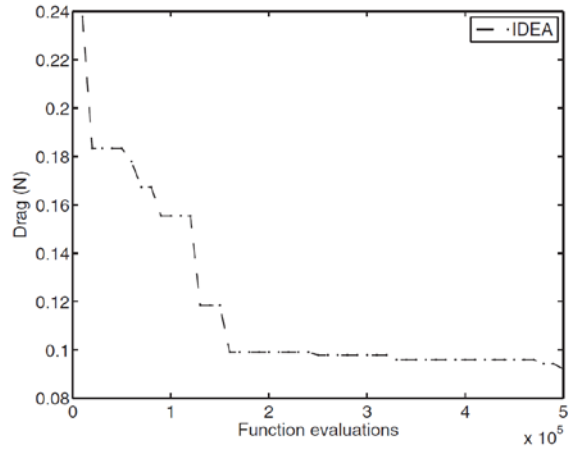


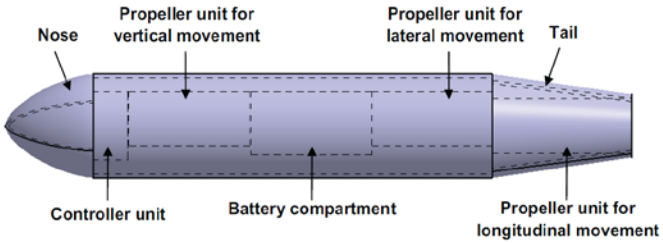
Fig. Progress plot of the best for robust design optimization

Approach 2: Results

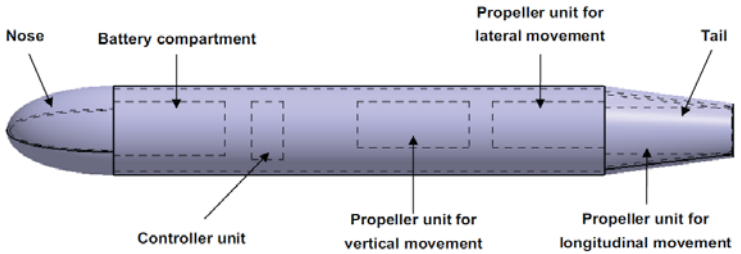
(a) Optimal solution

(b) Robust solution

Fig. CATIA models of the resulting toy submarines

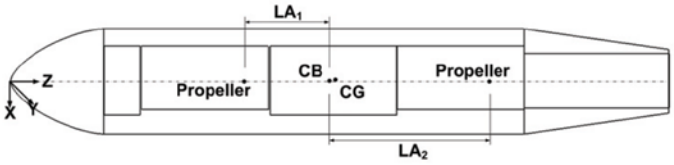


(a) Optimal solution

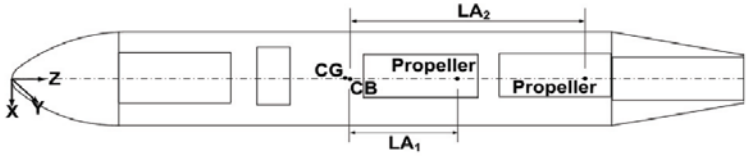


(b) Robust solution

Fig. Configurations of the resulting toy submarines



(a) Optimal solution



(b) Robust solution

Fig. Longitudinal sections of the resulting toy submarines

Approach 2: Results

Table: Comparison of optimization and robust solutions for perturbation of design variables

f		$f_{mean}(100)$	$f_{mean}(1000)$	N_g (100 func. evals.)			N_g (1000 func. evals.)		
				N_{g1}	N_{g2}	N_{g3}	N_{g1}	N_{g2}	N_{g3}
Optimum	0.082177 N	0.139804 N	0.137531 N	18%	48%	7%	11.9%	48.6%	6.2%
Robust	0.092112 N	0.093548 N	0.093492 N	0	0	0	0	0	0

- The results of the assessment for the robustness of the optimal and robust solutions for 100 and 1000 neighbours are reported in the above Table.
- It can be seen that the performance of the optimal solution degrades significantly, as the average drag values are 70% and 67% higher than the true optimal value for 100 and 1000 neighbours respectively.
- At the same time, the percentages of the number of violations (N_g) of each of the design constraints are noteworthy.
- On the other hand, the robust solution offers a better average performance and satisfies all the design constraints under the same perturbation, which is expected.

Approach 2: Results

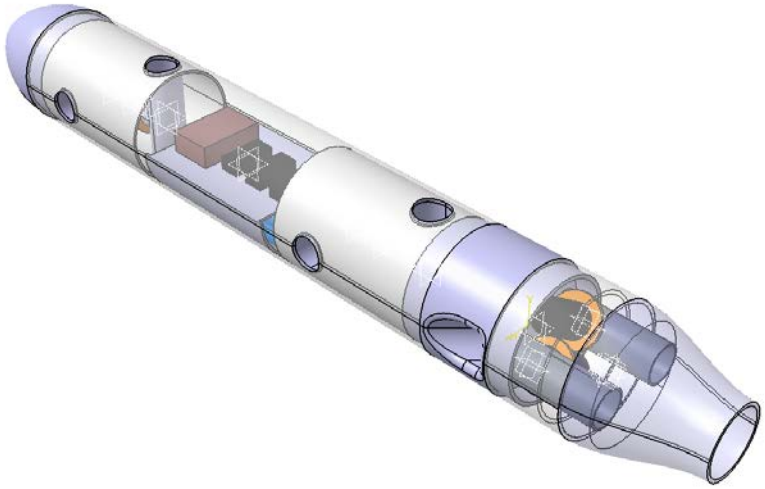
Table: Performance criteria of the existing USS Dallas RC toy, optimum and robust submarines

Vehicle particulars	USS Dallas	Optimum	Robust
Nose length	45 mm	41 mm	62 mm
Mid-body length	210 mm	231 mm	307 mm
Tail length	95 mm	80 mm	78 mm
Length overall	350 mm	360 mm	447 mm
Outer diameter	60 mm	58 mm	58 mm
L/D ratio	5.8	6.2	7.7
Wetted surface area	0.082385 m^2	0.082624 m^2	0.102018 m^2
Displacement volume	0.000437 m^3	0.000433 m^3	0.000598 m^3
Displaced water mass	437 g	433 g	598 g
Total mass of the vehicle	430 g	428 g	475 g
First lever arm length	45 mm	48 mm	71 mm
Second lever arm length	90 mm	90 mm	143 mm
CG-CB separation	3.705 mm	2.828 mm	2.171 mm
Nominal speed	0.5 m/s	0.5 m/s	0.5 m/s
Coefficient of viscous drag (MIT method)	0.008095	0.0079567	0.0072232
Drag (MIT method)	0.0825858 N	0.0821771 N	0.0921119 N

Design and Development of a Small, Low Cost UUV for Shallow Water Operations : White Sub

White Sub: Introduction

- Small UUV for shallow water operations
- Capable of supporting underwater communications research
- Relies heavily on the use of off-the-shelf components
- Modular configuration to adopt future enhancements
- Low cost

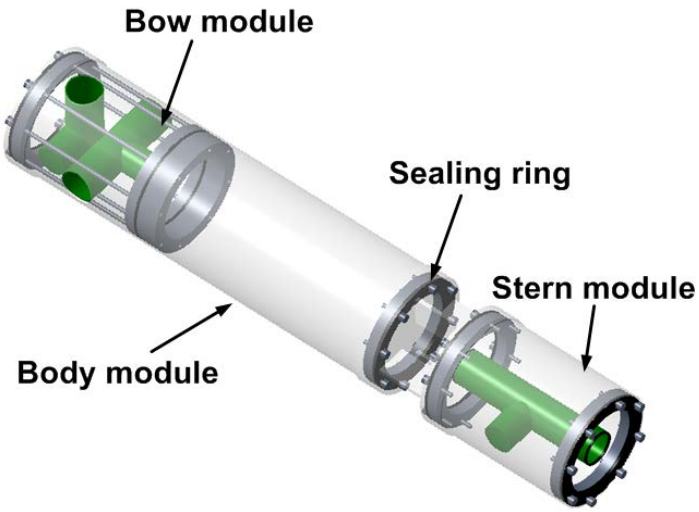


Gover, N., Hill, C., Alam, K., Ray, T., and Anavatti, S., "Design and development of a small, low-cost UUV for shallow water operations," In Proceedings of the 1st Submarine Institute of Australia Technology Conference, Adelaide, Australia, pp. 265-271, 2011.

White Sub: Design Requirements

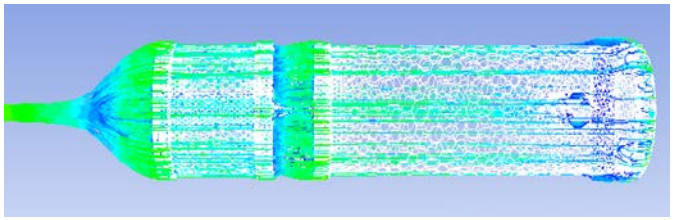
Design Specifications	
Performance	***Appearance***
6 Degree of freedom movements	Streamlined shaped
Maximum forward velocity of 1 m/s	No external appendages
Endurance of 20 mins at 1 m/s	Length overall to be less than 0.5 m
Range of 15 m	Fineness ratio (L/D) > 5
Remote controlled operations	Easily maintainable
Maximum operating depth of 5 m	Minimum watertight seals
Calm water operation	***Fabrication***
Maximum payload capacity of 1 kg	Production capability
Minimum payload compartment of 20X70X70 mm	Production of a prototype vehicle
Total weight to be less than 6 kg	***Budget***
Unspecified climb, dive and turn rates	Per unit < 1000 AUD

White Sub: Preliminary Design

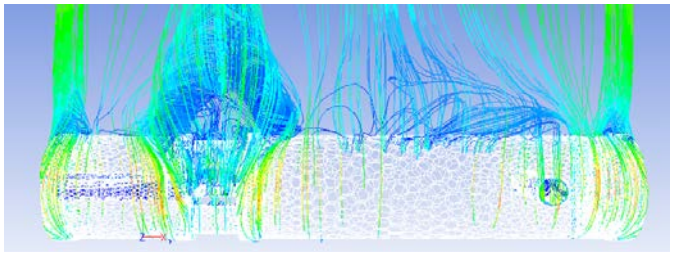


Vehicle Particulars			
Thickness (wall)	3.0 mm	Weight of the bow	0.607 kg
Outer diameter	110 mm	Weight of the body	0.689 kg
Inner diameter	104 mm	Weight of the stern	0.405 kg
Bow length	205 mm	Total weight	1.701 kg
Body length	255 mm	Total surface area	0.268 m^2
Stern length	204 mm	Total internal volume	0.002 m^3
Length overall	664 mm	Mass displaced (Fresh water)	2.094 kg
L/D ratio	6.04	Mass displaced (Salt water)	2.146 kg

White Sub: Performance Analysis of the Preliminary Design



(a) Forward motion



(b) Vertical motion

Fig. CFD analysis of the preliminary design

Flow Speed (m/s)	Forward Motion			Vertical Motion		
	Drag (N)			*Drag (N)*		
	Pressure	Viscous	Total	Pressure	Viscous	Total
0.5	1.9512	0.0718	2.0230	7.1932	0.2187	7.4119
1.0	7.8183	0.2245	**8.0428**	28.1286	0.1135	28.2421
1.5	17.6066	0.4427	18.0493	64.6425	0.6812	65.3237
2.0	31.3177	0.7191	32.0368	114.902	1.0949	115.997

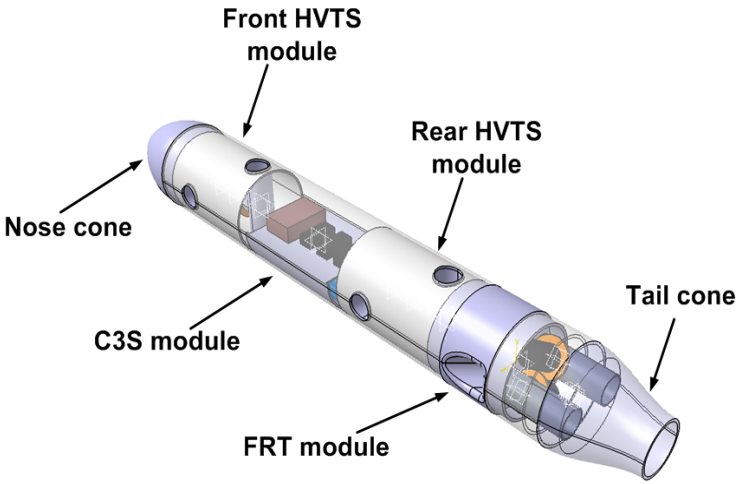
- Unable to meet the design specification of 1 m/s in forward motion, as the drag incurred by the vehicle is higher than that of the available thrust of 5 N.

White Sub: Performance of the Preliminary Design

MDO RESEARCH GROUP
Multidisciplinary Design Optimization

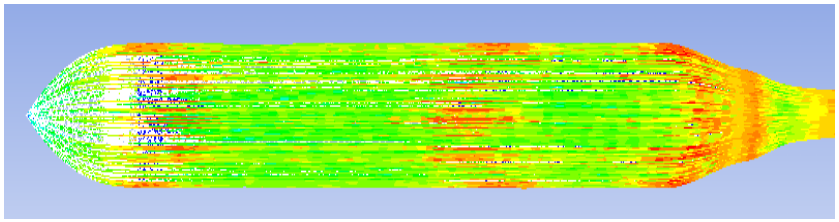
Design Specifications			
Performance		***Appearance***	
6 Degree of freedom movements	√	Streamlined shaped	√
Maximum forward velocity of 1 m/s	×	No external appendages	√
Endurance of 20 mins at 1 m/s	√	Length overall to be less than 0.5 m	×
Range of 15 m	√	Fineness ratio (L/D) > 5	√
Remote controlled operations	×	Easily maintainable	√
Maximum operating depth of 5 m	×	Minimum watertight seals	√
Calm water operation	×	***Fabrication***	
Maximum payload capacity of 1 kg	×	Production capability	√
Minimum payload compartment of 20X70X70 mm	√	Production of a prototype vehicle	×
Total weight to be less than 6 kg	√	***Budget***	
Unspecified climb, dive and turn rates	√	Per unit < 1000 AUD	√

White Sub: Detailed Design

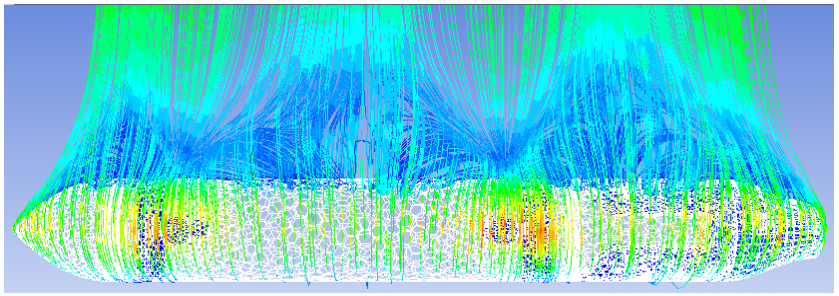


Vehicle Particulars			
Thickness (wall)	3.0 mm	Weight of the nose	0.308 kg
Outer diameter	110 mm	Weight of the PMB module	2.908 kg
Inner diameter	104 mm	Weight of the FRT module	1.056 kg
Nose cone	110 mm	Weight of the tail	0.310 kg
PMB module	600 mm	Total weight	4.582 kg
FRT module	151 mm	Total surface area	0.340 m^2
Tail cone	100 mm	Total internal volume	0.007 m^3
Length overall	961 mm	Mass displaced (Fresh water)	6.977 kg
L/D ratio	8.74	Mass displaced (Salt water)	7.151 kg

White Sub: Performance Analysis of the Detailed Design

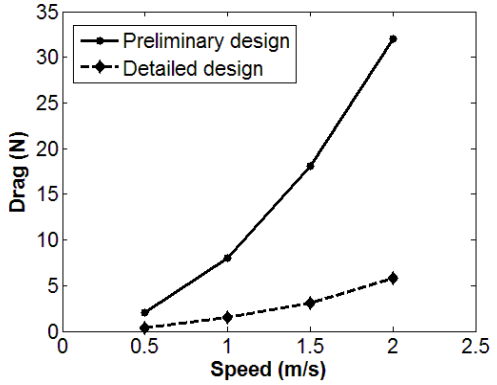


(a) Forward motion

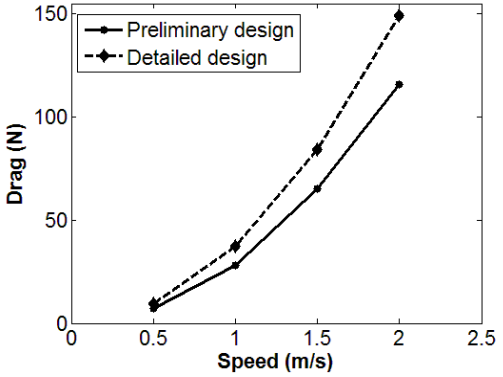


(b) Vertical motion

Fig. CFD analysis of the detailed design



(a) Forward motion



(b) Vertical motion

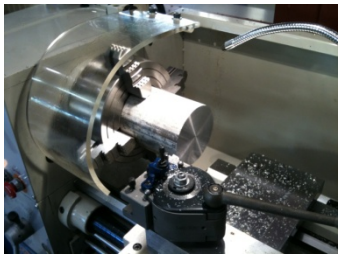
Fig. Comparison of drag between preliminary and detailed designs

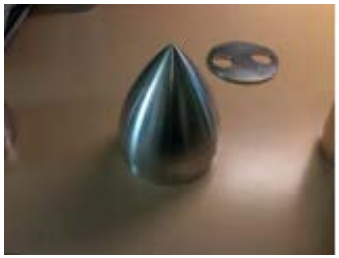
White Sub: Performance of the Detailed Design

Design Specifications			
Performance		***Appearance***	
6 Degree of freedom movements	√	Streamlined shaped	√
Maximum forward velocity of 1 m/s	√	No external appendages	√
Endurance of 20 mins at 1 m/s	√	Length overall to be less than 0.5 m	×
Range of 15 m	√	Fineness ratio (L/D) > 5	√
Remote controlled operations	√	Easily maintainable	√
Maximum operating depth of 5 m	√	Minimum watertight seals	√
Calm water operation	√	***Fabrication***	
Maximum payload capacity of 1 kg	√	Production capability	√
Minimum payload compartment of 20X70X70 mm	√	Production of a prototype vehicle	√
Total weight to be less than 6 kg	√	***Budget***	
Unspecified climb, dive and turn rates	√	Per unit < 1000 AUD	×

- The detailed design has met all the design specifications except length overall and budget. The overall length of the final design is 0.961 m and cost per unit is 2300 AUD.

White Sub: Fabrication

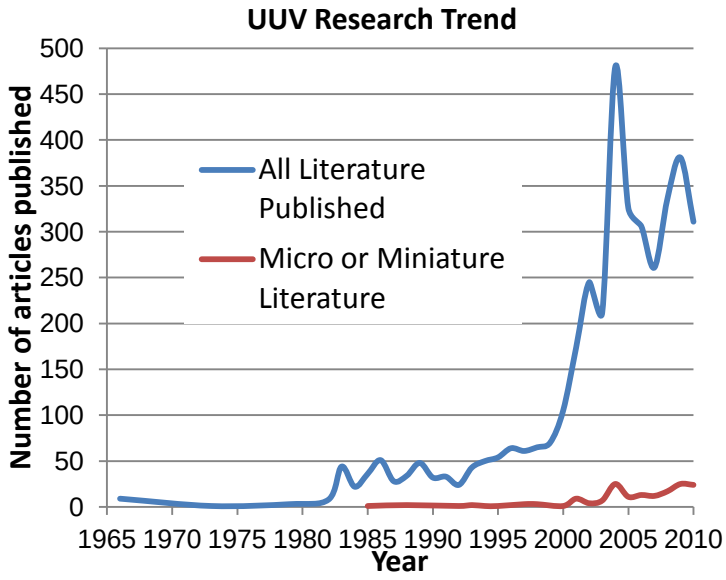





Design Optimization of a Micro Unmanned Underwater Vehicle (µUUV) : Six Inch Sub

Six Inch Sub: Introduction

- Limited attention has been paid towards micro UUV
- Micro UUVs are particularly attractive for deployment in extraordinarily confined spaces
- Can easily be transported to location and launched with little effort and minimum logistical support
- Multiple miniature vehicles could be deployed by a larger submersible to perform a task in collaboration
- Design optimization of a micro UUV to evaluate best design has not gain much attention by the researchers



Fearnley, J., and Ray, T., "Design and development of a six inch sub," In Proceedings of the 1st Submarine Institute of Australia Technology Conference, Adelaide, Australia, pp. 273-279, 2011.

Alam, K., Ray, T., and Anavatti, S., "Design optimization of a micro unmanned underwater vehicle (μUUV)," School of Engineering and Information Technology, UNSW Canberra, 2013.

Six Inch Sub: Design Requirements

- Operating speed of the UUV should be 0.2 m/s
- Length of the UUV must be no more than 152.4 mm (6 inch)
- Should be able to house a camera of 2.87E4 mm^3 volume and radius of 19 mm
- The vehicle is to be propelled by one rear propeller and two propellers for yaw movement. The pitch movement will be achieved through linear actuator and syringe mechanism
- Should have enough space to carry a controller and a battery unit
- The UUV must be able to store lead shot for ballast and weight in order to balance the buoyant force for underwater use
- Should maintain a modular configuration for easy access to the internal components and be reconfigured to suit various mission requirements

Six Inch Sub: Hull Geometry

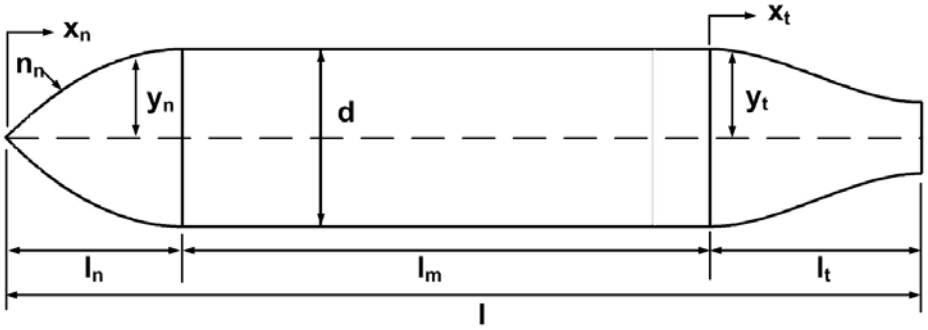


Fig. Parameterization of the hull geometry

- Equation of the nose:

$$y_n = \frac{1}{2} d \left[1 - \left(\frac{l_n - x_n}{l_n} \right)^{n_n} \right]^{\frac{1}{n_n}}$$

where y_n is the radius of the nose, d is the maximum body diameter, which may be varied, l_n is the length of the nose, x_n is the reference length that varies from 0 to l_n, and n_n is the shape variation coefficient of the nose which may also be varied to give different shapes of the nose.

- Equation of the tail:

$$y_t = 0.0002 x_t^{\,3} - 0.0237 x_t^{\,2} + \frac{d}{2}$$

where y_t is the radius of the tail and x_t is the reference length that varies from 0 to l_t.

Six Inch Sub: Internal On-board Components

- The submarine employs a propeller for surge control, two small propellers rotating about the vertical axis for yaw control and a hydrostatic displacement changing system for vertical displacement.
- The AAA size NiMH batteries with nominal cell voltage of 3.6 V are used as the power source.
- The image capturing capability of the UUV will be achieved by a standalone digital video recording (DVR) ‘spy’ camera which will be positioned forward most for visibility.

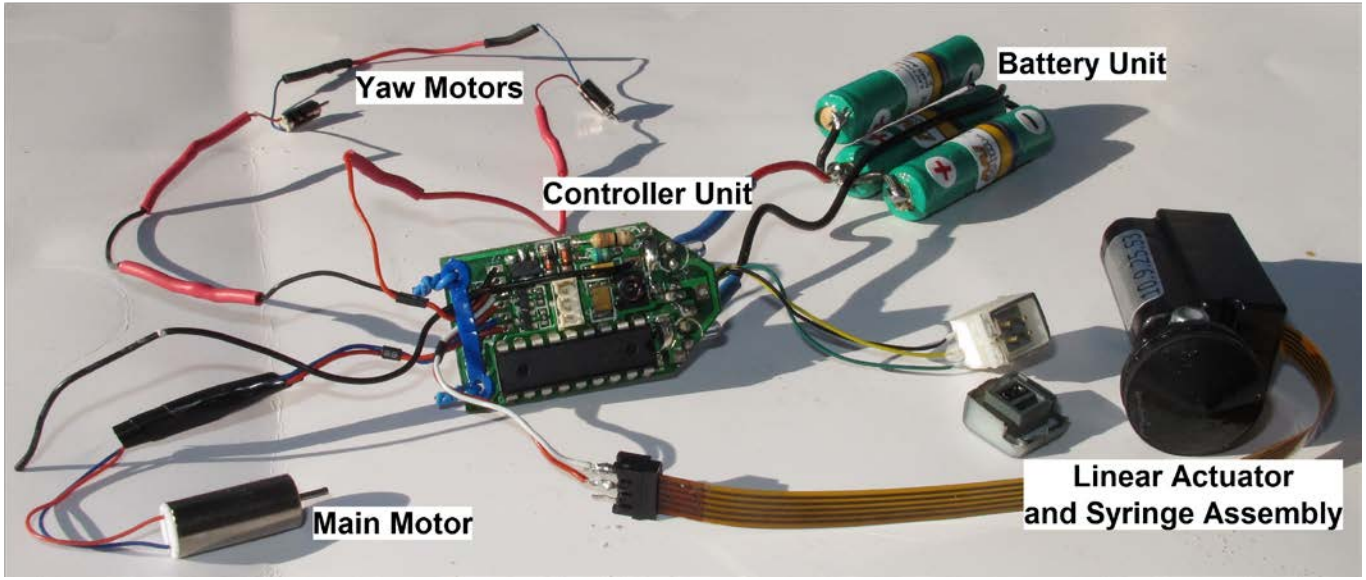


Fig. Internal on-board components to be used for the vehicle design

Six Inch Sub: Hydrostatic Displacement Changing System

- The displacement changing system utilizes a geared stepper motor-actuated syringe mechanism which adjusts the displacement of the submarine.
- The syringe cavity is designed to run horizontally and positioned as far forward as possible without clashing with the camera to increase the induced pitch.
- The cavity is considered as half full at the state of neutral buoyancy. When the syringe is extended, the face of the syringe is flush with the hull of the submarine and this state gives positive buoyancy.
- As the syringe face is brought back inside the submarine, the vehicle's displacement decreases and its buoyancy consequently decreases giving negative buoyancy.
- This change of attitude means that when the submarine tries to ascend or descend, the camera can see the direction in which it is heading and also the thrust can be used to increase the rate of ascent.

Fig. Micro eyes ball camera

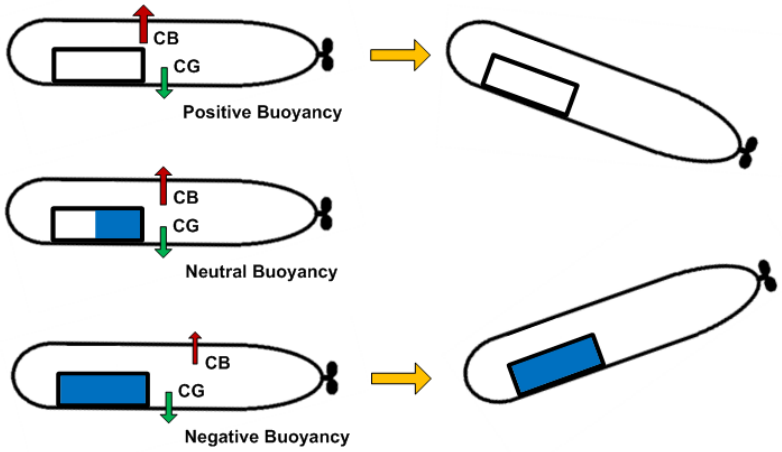


Fig. Hydrostatic displacement changing system induced pitch schematic

Six Inch Sub: Problem Formulation

Minimize :

$f(1) = D$

Subject to :

$g(1) = l \le 152.4\,mm;\ g(2) = l/d \ge 2.8$

$g(3) = LA_1 \ge 35\,mm;\ g(4) = LA_2 \ge 35\,mm$

$g(5) = s \le 4\,mm;\ g(6) = ID_n \ge 38\,mm$

$g(7) = IV_n \ge 2.8742E4\,mm^3$

Variable bounds :

$0 \le X_A \le 50\,mm;\ 0 \le Y_A \le 50\,mm;\ 0 \le Z_A \le 50\,mm$

$0 \le X_B \le 50\,mm;\ 0 \le Y_B \le 50\,mm;\ 0 \le Z_B \le 50\,mm$

$0 \le X_C \le 50\,mm;\ 0 \le Y_C \le 50\,mm;\ 0 \le Z_C \le 50\,mm$

$0 \le X_W \le 50\,mm;\ 0 \le Y_W \le 50\,mm;\ 0 \le Z_W \le 50\,mm$

$0 \le X_{Y1} \le 50\,mm;\ 0 \le Y_{Y1} \le 50\,mm;\ 0 \le Z_{Y1} \le 50\,mm$

$0 \le X_{Y2} \le 50\,mm;\ 0 \le Y_{Y2} \le 50\,mm;\ 0 \le Z_{Y2} \le 50\,mm$

$2 \le n_n \le 3;\ 40 \le l_n \le 45\ mm$

$30 \le l_t \le 35\,mm$

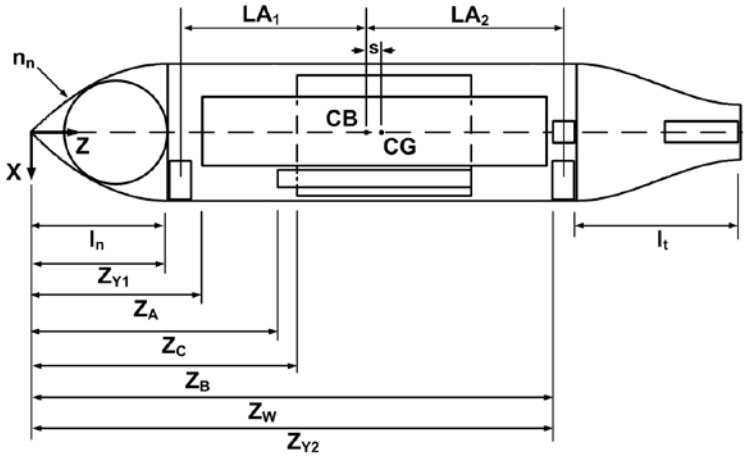


Fig. The constraints and design variables for problem formulation

Six Inch Sub: Results

Table: Single objective drag minimization results

Drag	IDEA
Best	0.008178 N
Mean	0.008178 N
Median	0.008178 N
Worst	0.008183 N
SD	9.12871E-7 N

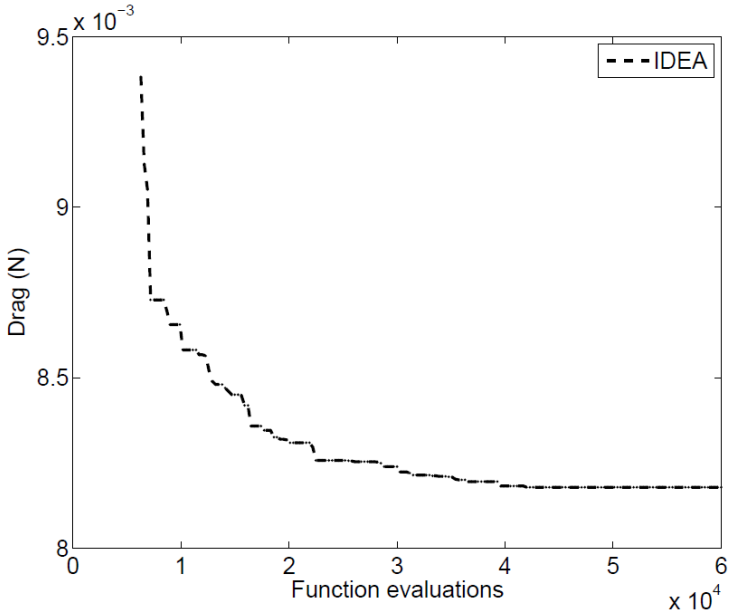


Fig. Progress plot of the best design for single objective drag minimization using IDEA

Six Inch Sub: Results

Fig. Bare hull of the best design

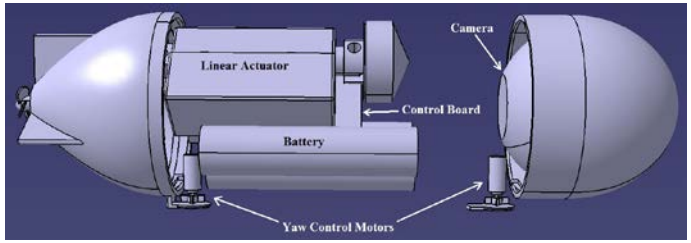
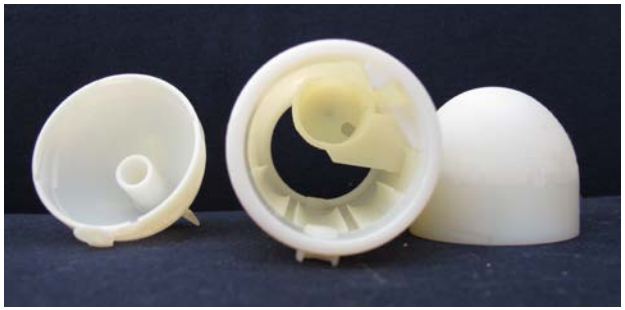


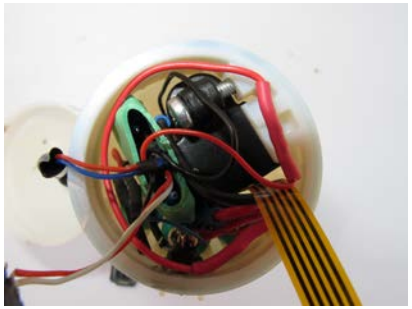
Fig. Internal image of the final design

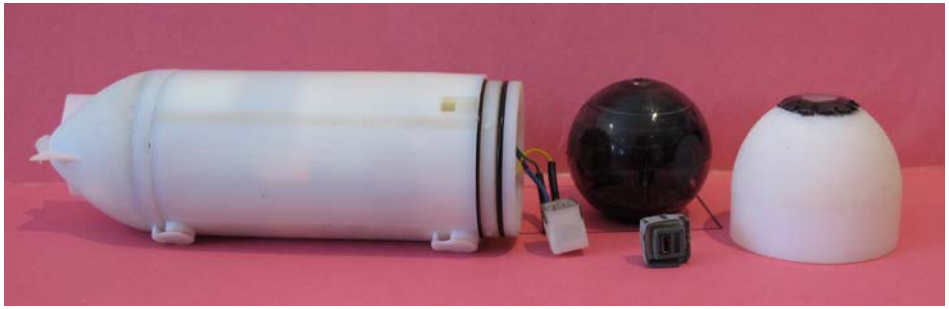
Table: Performance criteria of the optimized six inch sub

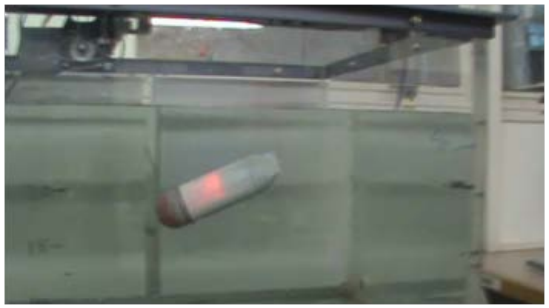
Vehicle Particulars			
Nose length	40 mm	Total mass of the vehicle	172.17 g
Mid-body length	78 mm	Max. inner square size	34.02 mm
Tail length	34.4 mm	First lever arm length	37.44 mm
Length overall	152.4 mm	Second lever arm length	38.41 mm
Outer diameter	51.1 mm	CG-CB separation	3.88 mm
L/D ratio	2.98	Nominal speed	0.2 m/s
Wetted surface area	0.021611 m^2	Drag (VT method)	0.006338 N
Displacement volume	0.000247 m^3	Drag (G&J method)	0.006733 N
Displaced water mass	0.247 kg	Drag (MIT method)	0.008178 N

Six Inch Sub: Fabrication







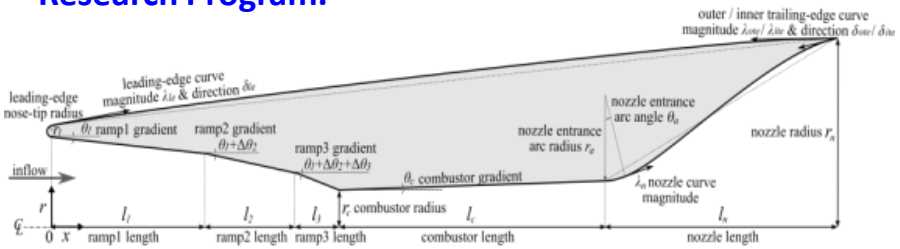


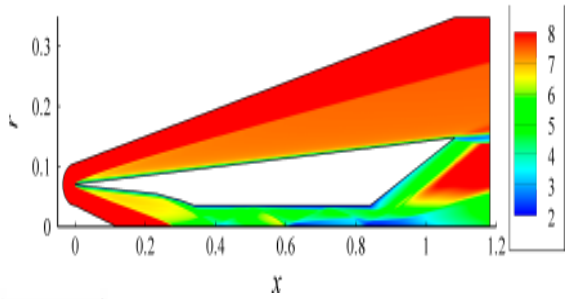
Benefits of the Framework

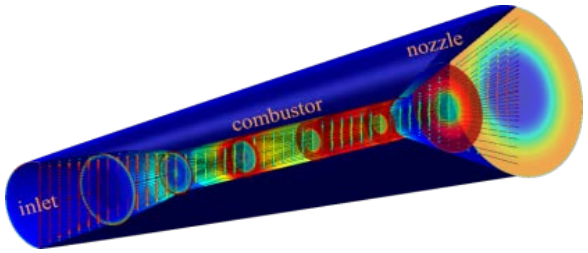
- *Novelty:* The framework is the first of its kind to offer full multidisciplinary design optimization functionalities to be considered in the design of UUVs.
- *Modularity:* The modular design optimization framework is reconfigurable to suit various design requirements.
- *Robustness:* To deliver practical designs, the formulations have been extended to yield *robust* optimal designs.
- *Interfacing:* Seamless integration of Matlab-CATIA-ANSYS (ICEM, FLUENT) and in-house performance analysis codes.
- The framework has been used to design and build a number of UUVs to support underwater communications research.

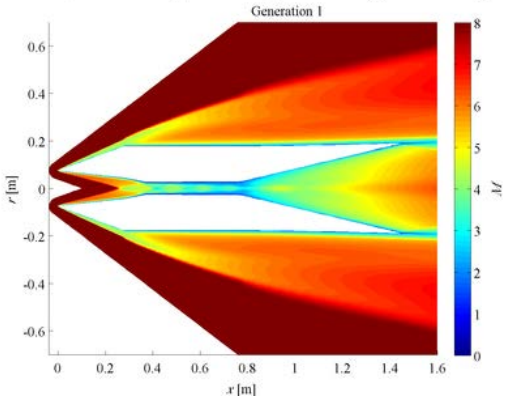
Application Snapshot: Scramjet Shape

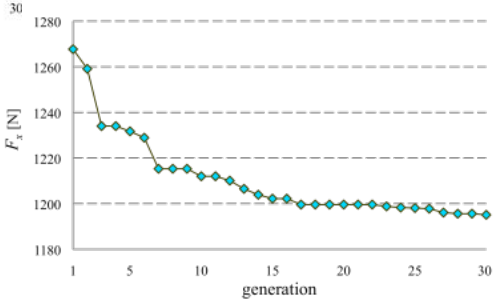
Full Flow-Path Design Optimization and Analysis of Axisymmetric Scramjets. Funded by Australian Space Research Program.

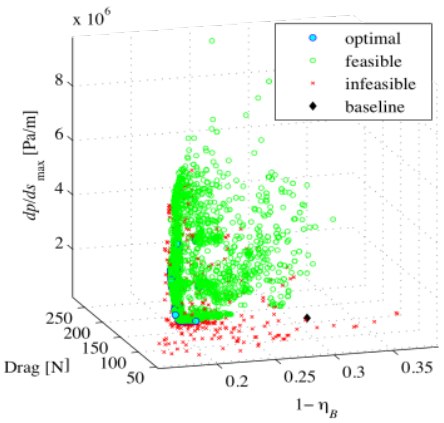


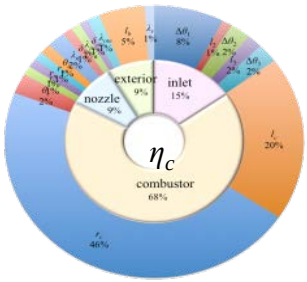












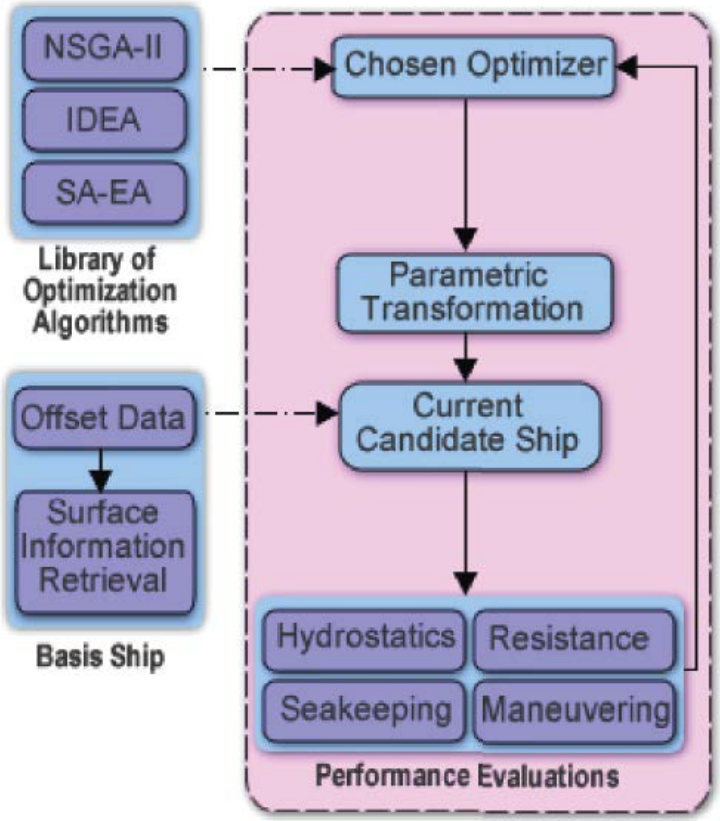
SCRAMSPACE

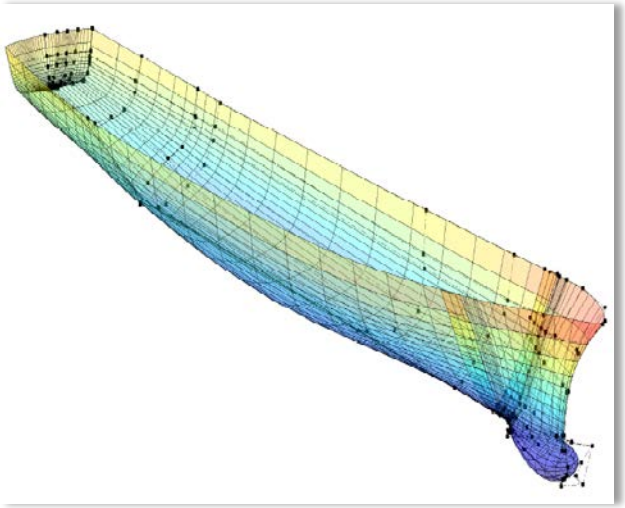
UQ

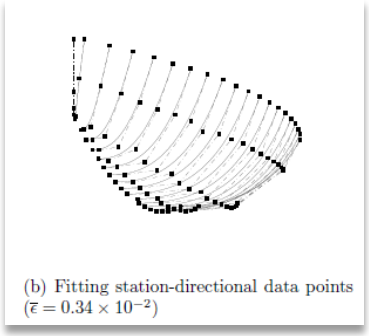
Ogawa,H. Brown,L. Boyce,R.R., and Ray,T., Multiobjective Design Optimization of Axisymmetric Scramjet Nozzle and External Components Considering Static Stability by using Surrogate Assisted Evolutionary Algorithms, International Society of Air-breathing Engines, ISABE 2011, September 12-16, 2011 Gothenburg, Sweden.

Application Snapshot: Planning Craft

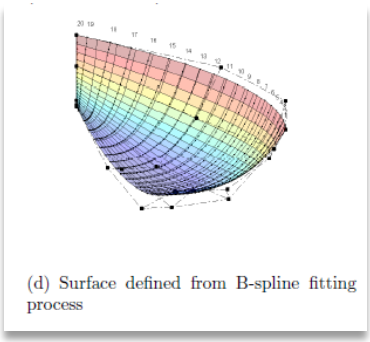
Framework for Design Optimization of High Speed Craft







(b) Fitting station-directional data points ($\bar{\epsilon} = 0.34 \times 10^{-2}$)



(d) Surface defined from B-spline fitting process

1. A library of optimization algorithms
2. Surface information retrieval modules
3. Parametric transformation module
4. Hydrostatics and hydrodynamics module

Mohamad, A.F.A., Ray, T., and Smith, W.(2011), Uncovering Secrets Behind Low Resistance Planing Craft Hull Forms Through Optimization, *Engineering Optimization*.iFirst, 2011, pp. 1-13.

Shape Representation

Non-uniform rational b-splines (NURBS) has been acknowledged as a well accepted standards in surface representation.

Its simple, elegant surface representation are now a norm among various CAD tools e.g. AutoCAD, Maxsurf, Friendship Framework etc.

Such an approach requires: Control points, Knot vectors, Order of the curve/surface and the Basis function.

Mathematically it can be represented as $[D] = [N]\,[Q]$

where D is the resulting surface, N is the basis function and Q denotes the control points.

$$[D] = [N]\,[Q]$$

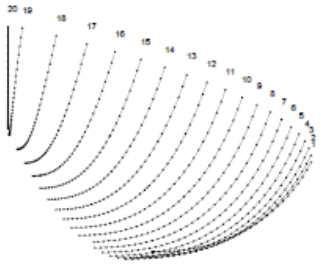
$$[N]^T\,[D] = [N]^T\,[N]\,[Q]$$

$$[Q] = \left[[N]^T\,[N]\right]^{-1}\,[N]^T\,[D]$$

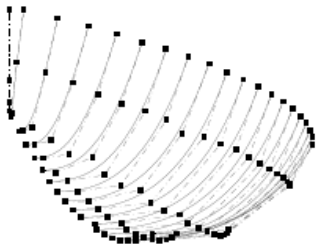
$$\epsilon = \sqrt{(Q_x - S_x)^2 + (Q_y - S_y)^2 + (Q_z - S_z)^2}$$

The fitting error can be estimated as above.

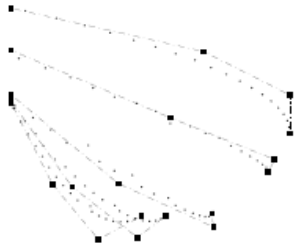
Shape Representation



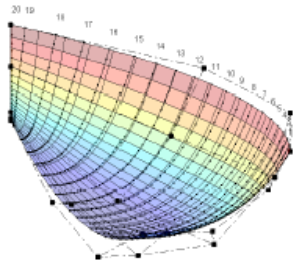
(a) Data points arranged in numbered stations offsets



(b) Fitting station-directional data points ($\overline{\epsilon} = 0.34 \times 10^{-2}$)



(c) Fitting longitudinal direction through control points of station-directional control points ($\overline{\epsilon} = 0.15$)

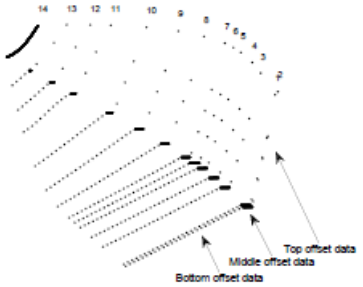


(d) Surface defined from B-spline fitting process

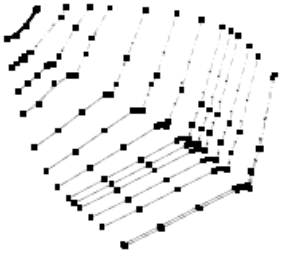
Fitting the surface all at once

Mohamad, A.F.A. , Ray, T. , and Smith, W. , "Uncovering secrets behind low resistance planing craft hull forms through optimization," Engineering Optimization, vol. 43, no. 11, November, pp. 1161–1173, 2011

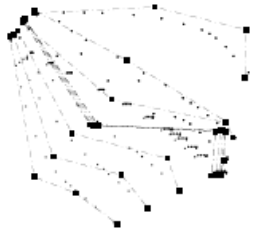
Shape Representation



(a) Original station-based data points

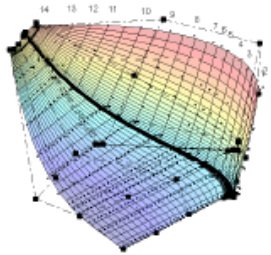


(b) Fitting station-directional data points ($\overline{\epsilon}_{\text{top}} = 0.92 \times 10^{-2}$, $\overline{\epsilon}_{\text{middle}} = 0.78 \times 10^{-9}$, $\overline{\epsilon}_{\text{bottom}} = 0.95 \times 10^{-8}$)



(c) Longitudinal control points and corresponding B-spline ($\overline{\epsilon}_{\text{top}} = 0.67$, $\overline{\epsilon}_{\text{middle}} = 0.67$, $\overline{\epsilon}_{\text{bottom}} = 0.48$)

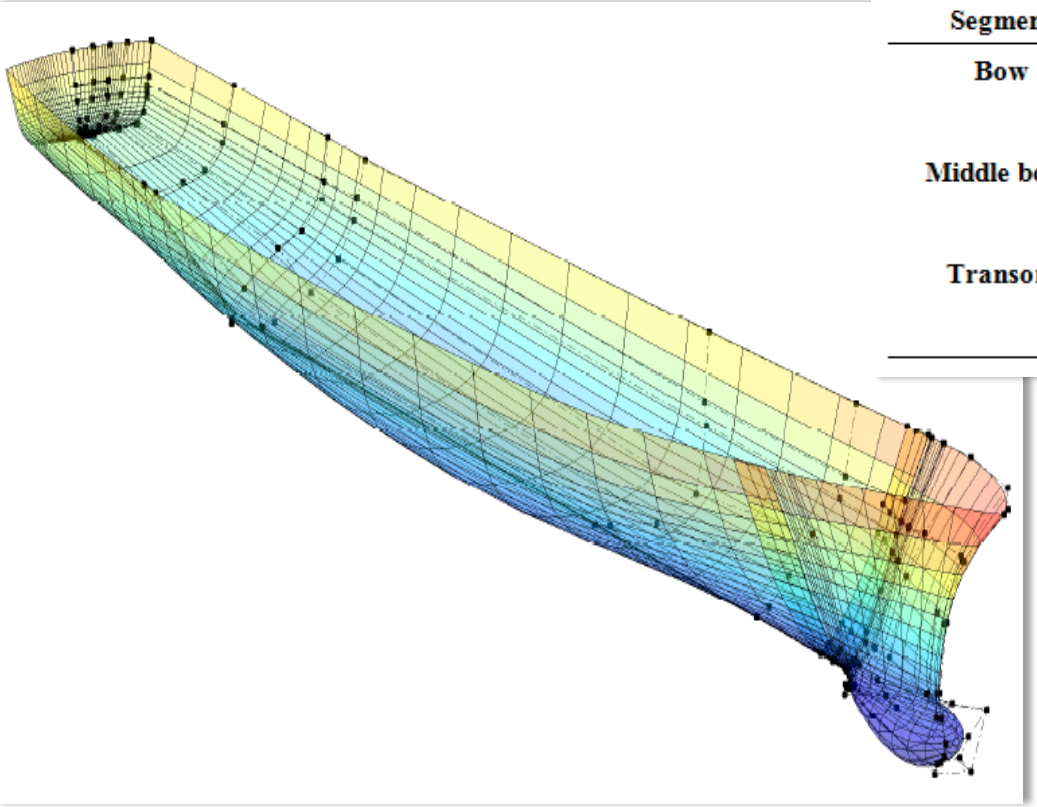
Fitting the multiple surfaces., especially when there is a chine(discontinuity).



(d) Surface defined from longitudinal net

Mohamad, A.F.A. , Ray, T. , and Smith, W. , "A hydrodynamic preliminary design optimization framework for high speed planing craft," Journal of Ship Research, vol. 56, No. 1, pp. 35–47, 2012.

Shape Representation



US Navy DTMB 5415 fitting error

Segment	Error Description	Error Value
Bow	$\bar{\varepsilon}_{\text{station}}$	4.5×10^{-3}
	$\bar{\varepsilon}_{\text{longitudinal}}$	4.3×10^{-3}
Middle body	$\bar{\varepsilon}_{\text{station}}$	4.6×10^{-8}
	$\bar{\varepsilon}_{\text{longitudinal}}$	2.3×10^{-3}
Transom	$\bar{\varepsilon}_{\text{station}}$	5.4214×10^{-7}
	$\bar{\varepsilon}_{\text{longitudinal}}$	5.6330×10^{-7}

Mohamad, A.F.A. , Ray, T. , and Smith, W. , “Beyond hydrodynamic design optimization of planing craft,” Journal of Ship Production and Design, vol. 27, no. 1, pp. 1–13, 2011.

Surface information retrieval of a DTMB 5415 (destroyer craft)

Planing Craft Design: Case Study 1

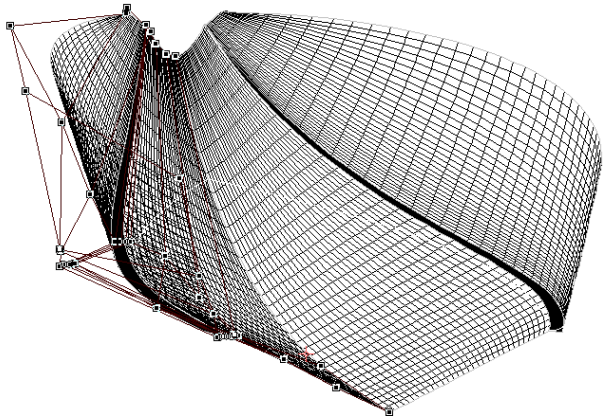
Background of the planing craft

The planing craft used in this study represents a craft similar to U.S. Coast Guard (USCG) Surf Rescue Boat (30-foot SRB) (*Halberstadt (1987)*). The ship is designed to operate in sea up to 3 m waves, with a maximum speed of 30 knots.

Characteristics of the planing craft

Displacement	7204.94 kg
Length	10.04 m
Beam	2.86 m
Draft	0.7 m
Metacentric height (*GM*)	2.0 m
Speed	20.81 kts
$C_V = V/(g \times B)^{1/2}$	2.02

USCG 30 foot SRB *Halberstadt (1987)*)



Basis hull used in the case study

Planing Craft Design

Problem formulation

- The optimization problem is posed as the identification of a planing craft with minimum total resistance subject to the constraints on displacement, stability (transverse metacentric height) and impact acceleration corresponding to the operational sea-states.

Minimize:
$f = R_T$, where $R_T = R_C + R_A$

Design variables:
$9m < L < 11m$ (L_B=10.04m) ;
$1.8m < B < 3.8m$ (B_B=2.862m) ;
$0.6m < T < 0.8m$ (T_B=0.7m)

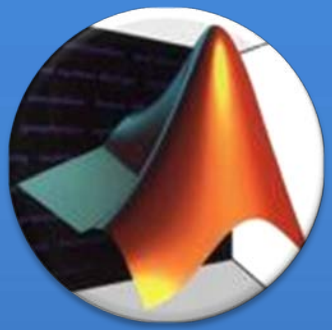
Constraints:
$g(1) : Disp_I > Disp_B$; $g(2) : GM_I \geq GM_B$;
$g(3) : 3.07 < L_I/Vol_I^{1/3} < 12.4$; $g(4) : 3.7^o < I_{el} < 28.6^o$;
$g(5) : 2.52 < L_I/B_I < 18.28$; $g(6) : 1.7 < B_I/T_I < 9.8$;

Nomenclature:
R_T = Total resistance
R_C = Calm water resistance
R_A = Added resistance due to waves
L = Length
B = Beam
T = Draft
$Disp$= Displacement
GM = Transverse metacentric height
Vol = Displaced volume
I_{el} = Half angle of entrance

Subscript I: Candidate design
Subscript B: Basis design

For each algorithm (NSGA-II, IDEA and EASDS), 10 independent runs are performed. A population size of 40, crossover probability of 1, mutation probability of 0.1, crossover distribution index of 10, and mutation distribution index of 20 were used for each algorithm. The number of function evaluations used by each algorithm is kept approximately equal for a fair comparison. The surrogate models used are restricted to RSM, ORSM, RBF, ORBF and DACE. A training period of 3 and prediction error of 0.05 has been used for EASDS.

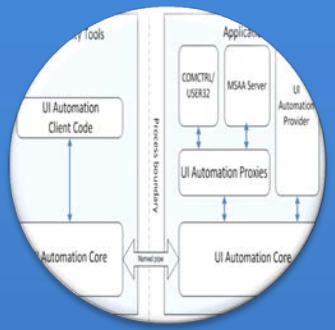
- **Mathematical expressions** & **design tools** for the planing craft design optimization framework:
  - Hydrostatics (displacement, stability)
  - *Savitsky (1964)* regression equations for calm water resistance estimation
  - *Savitsky and Ward Brown (1976), Savitsky and Koelbel (1993)* expressions derived from *Fridsma's (1971)* experimental tank test data on planing craft operating in rough water for added resistance due to waves and vertical impact acceleration estimation.
  - *Lackenby (1950)* and *Maxsurf (2007)* parametric transformation method.
  - *Mohamad Ayob et al. (2009)* surface information module, based on the inverse B-spline method, *Rogers and Adams (1990)*



Matlab (The Mathworks)

FORMSYS

Maxsurf (Formsys Inc.)



Microsoft COM Automation

Incorporation of Commercial Codes / Analysis Tools

Planing Craft Design

- A hydrodynamic optimization framework for a hard chine planing craft in seaway operations is presented in this work. The proposed framework incorporates three evolutionary algorithms, namely **NSGA-II, EASDS and IDEA**.

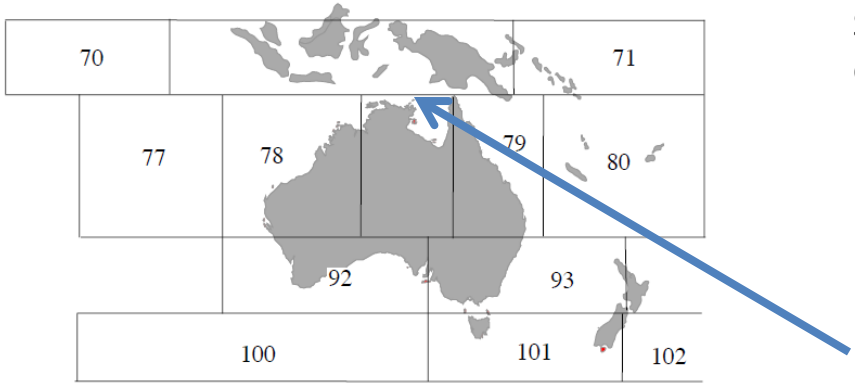
- The hull form optimization problem is formulated through minimization of R_T in four sea-states, with $H_{1/3}$ of 0.4m, 0.6m, 0.8m and 1.1m **with and without vertical impact acceleration constraints** to illustrate scenarios for **manned and unmanned missions**.

- The framework allows an easy integration of various analysis modules of varying fidelity. The ability to generate an optimum hull form rather than the optimum principal dimensions allows for a better estimate of performance, while at the same time providing offsets directly to support other detailed analysis and even direct construction.

- The inclusion of **surrogate models through EASDS** allows the possibility to identify better designs for the same computational cost as highlighted in the case studies. The proposal to **accelerate the rate of convergence through the use of IDEA** for constrained optimization problems is also illustrated. The importance and effects of the impact acceleration constraint on manned and unmanned missions are discussed.

- The proposed framework being modular in nature, allows for the possibility of including other underlying optimization schemes or high fidelity multidisciplinary analysis tools to support design of hard chine planing crafts.

Coastal surveillance scenario setting



Hogben, N., Dacunha, N. M. C. and Olliver, G. F.: Global Wave Statistics: compiled and edited by British Maritime Technology, British Maritime Technology, 661 pp. (1986).

Sea-states based on Pierson-Moskovitz energy spectrum can be derived from wind speed data on a particular coastal area or available recorded data.

Sea-state	$H_{1/3}$ (m)
Sea-state 3	1.34
Sea-state 4	2.32
Sea-state 5	3.62

- Design the vessel for minimum total resistance at Sea states 3, 4 and 5.
- The boxes surrounding Australian map are the 'Definition of Areas' by Hogben *et. al.*, (1986).
- It is assumed that the ship will operates in northern Australian waters. Realistic estimation can be made form the work of Hogben *et. al.*, (1986).

Planing Craft Design: Performance if WPB110

The performance of WPB110 in sea-states 3, 4, 5 are as follows:

	Sea –state 3	Sea –state 4	Sea –state 5
Displacement (N)	118246.596	118246.597	118246.596
LWL (m)	32.930	32.930	32.930
Beam (m)	7.518	7.518	7.518
Draft (m)	1.600	1.600	1.600
GMt (m)	1.638	1.638	1.638
Calm Water Resistance (N)	135932.36	135932.36	135932.36
Deadrise Angle	31.860	31.860	31.860
Running Trim	3.760	3.760	3.760
Speed (knots)	30	30	30
Sig Wave Height	1.340	2.320	3.620
Added Resistance (N)	40700.979	52414.029	65498.889
Impact Acc. (g)	0.754	1.128	1.625
Total Resistance (N)	176633.339	188346.389	201431.249

Is it possible to design better forms than WPB110 for Sea states of 3, 4 and 5 ?

Best design across ten runs

		NSGA-II	EASDS	IDEA
Sea State 3	Avg. Num. of Fun.Eval.	720	688.4	720
	% of Minimized Resistance	7.35	7.71	**7.74**
Sea State 4	Avg. Num. of Fun.Eval.	880	856.6	880
	% of Minimized Resistance	5.69	5.94	**6.11**
Sea State 5	Avg. Num. of Fun.Eval.	760	744.3	760
	% of Minimized Resistance	**4.19**	4.02	3.99

Median design across ten runs

		NSGA-II	EASDS	IDEA
Sea State 3	Avg. Num. of Fun. Eval.	720	688.4	720
	% of Minimized Resistance	6.96	7.37	**7.39**
Sea State 4	Avg. Num. of Fun.Eval.	880	856.6	880
	% of Minimized Resistance	5.08	4.61	**5.53**
Sea State 5	Avg. Num. of Fun.Eval.	760	744.3	760
	% of Minimized Resistance	3.41	**3.89**	3.63

Worst design across ten runs

		NSGA-II	EASDS	IDEA
Sea State 3	Avg. Num. of Fun.Eval.	720	688.4	720
	% of Minimized Resistance	5.59	5.81	**6.35**
Sea State 4	Avg. Num. of Fun.Eval.	880	856.6	880
	% of Minimized Resistance	2.90	3.88	**4.58**
Sea State 5	Avg. Num. of Fun.Eval.	760	744.3	760
	% of Minimized Resistance	2.51	**2.77**	2.71

Why Identifying an Optimum Design based on Calm Water Resistance is Not Enough ?

Ship optimized at calm water only VS ship optimized at Sea State 4 (EASDS results)

	OPT @ Calm	% reduction	OPT @ SS4	% reduction
Analysed @ Calm Water (N)	122488.78	9.89	122600.45	9.81
Analysed @ SS3 (N)	163684.45	7.33	162633.71	7.93
Analysed @ SS4 (N)	178663.11	5.14	177165.88	5.94
Analysed @ SS5 (N)	195514.33	2.94	193515.58	3.93

Each optimized ship is compared with the respective basis ship operating at the respective sea-states

A significant difference is observed between ship optimized in calm water only and ship optimized in sea-state condition.

Best design across ten runs

		NSGA-II	EASDS	IDEA
Sea State 3	Avg. Num. of Fun.Eval.	720	688.4	720
	% of Minimized Resistance	7.35	7.71	**7.74**
Sea S		880	856.6	880
		5.69	5.94	**6.11**
Sea S		760	744.3	760
	% of Minimized Resistance	**4.19**	4.02	3.99

Given the same approximate number of function evaluations, IDEA consistently reports better percentage of reduction in all runs

Median design across ten runs

		NSGA-II	EASDS	IDEA
Sea State 3	Avg. Num. of Fun.Eval.	720	688.4	720
	% of Minimized Resistance	6.96	7.37	**7.39**
Sea S		880	856.6	880
	% of Minimized Resistance	5.08	4.61	**5.53**
Sea State 5	Avg. Num. of Fun.Eval.	760	744.3	760
	% of Minimized Resistance	3.41	**3.89**	3.63

However, less than 10% of reduction can be achieved in this case study.

Wo

		NSGA-II	EASDS	IDEA
Sea State 3	Avg. Num. of Fun.Eval.	720	688.4	720
	% of Minimized Resistance	5.59	5.81	**6.35**
Sea State 4	Avg. Num. of Fun.Eval.	880	856.6	880
	% of Minimized Resistance	2.90	3.88	**4.58**
Sea State 5	Avg. Num. of Fun.Eval.	760	744.3	760
	% of Minimized Resistance	2.51	**2.77**	2.71

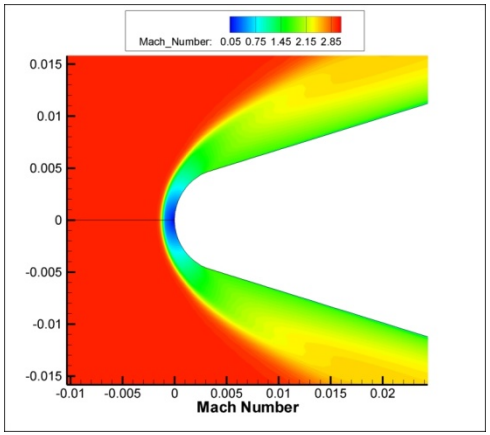
This case study might suggest that the design of a typical coast guard planing craft could have been well designed for seaway operation.

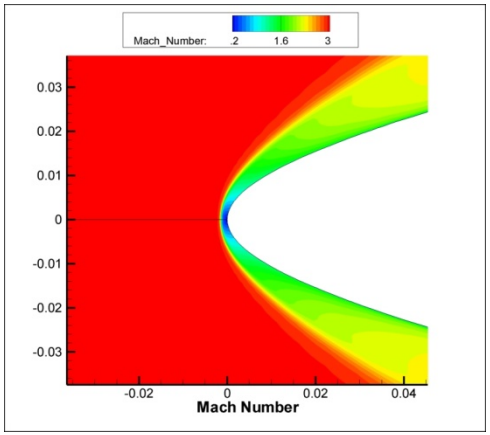
Application Snapshot: Miscellaneous

Nose Cone Shape Optimization

Approach: With and without surrogates

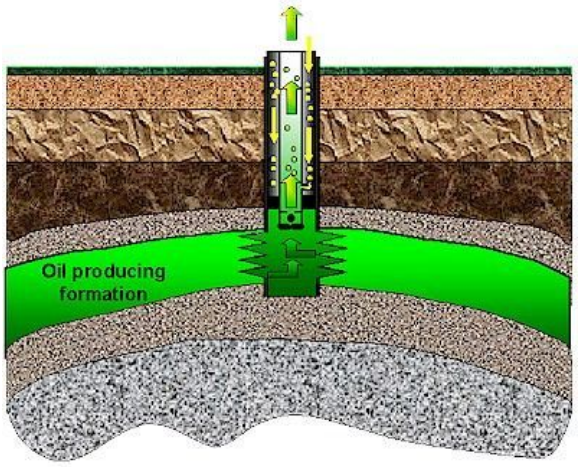
	M = 3.02	M = 8.04
Base Design	0.2601	0.3001
Without Surrogate	0.2565 (1.39%)	0.2942 (1.96%)
With Surrogate	0.2569 (1.21%)	0.2977 (0.80%)
Computational Saving	18%	11%





Deepak, R., Ray. T. and Boyce, R. Evolutionary Algorithm Shape Optimization of a Hypersonic Flight Experiment Nose Cone, Journal of Spacecrafts and Rockets, 45 (3), pp. 428-437,2008.

Optimal Gas Injection Planning



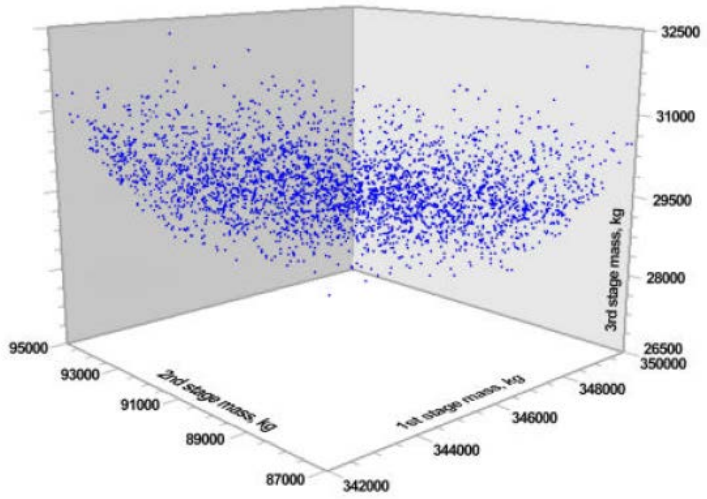
Six Well		Fifty Six Well	
Buitrago et al.	3629.00	Buitrago et al.	21789.9
Best EA	3663.99	Best EA	22033.4
Worst EA	3653.90	Worst EA	21222.4
Average EA	3660.20	Average EA	21622.3
Median EA	3660.77	Median EA	21651.2

An Increase of 243 Barrels per Day for 56 Oil Well Problem (Benchmark Problem)

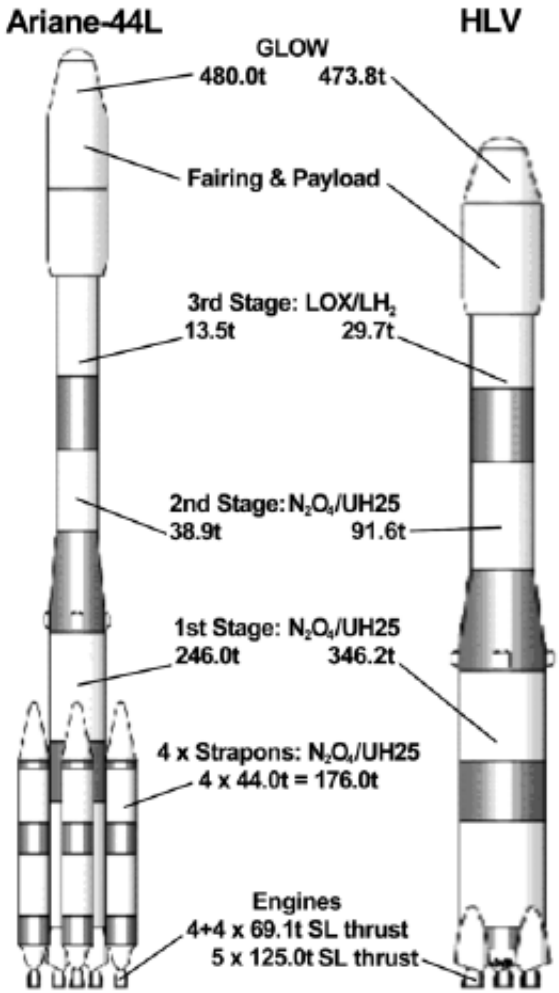
Ray,T. and Sarker, R. Genetic Algorithm for Solving a Gas Lift Optimization Problem, *Journal of Petroleum Science and Engineering,* Vol. 59, pp. 84-96, 2007.

Ray,T. and Sarker,R., Evolutionary Algorithms Deliver Promising Results to Gas Lift Optimization Problems, World Oil, April 229 (4), pp. 141-142, 2008.

Launch Vehicle Design



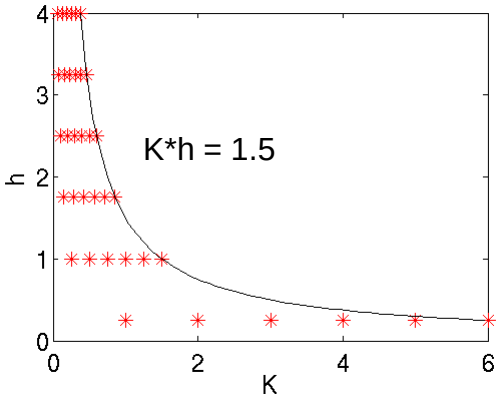
47 Kg Reduction in Total Stage Masses



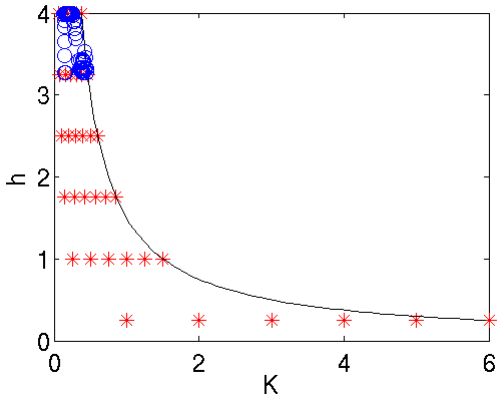
Briggs, G.P., Ray, T. and Milthorpe, J.(2007). Optimal Design of an Australian Medium Launch Vehicle, *Innovations in Systems and Software Engineering*, (A NASA Journal), Springer. Vol. 3, pp. 105-116,2007.

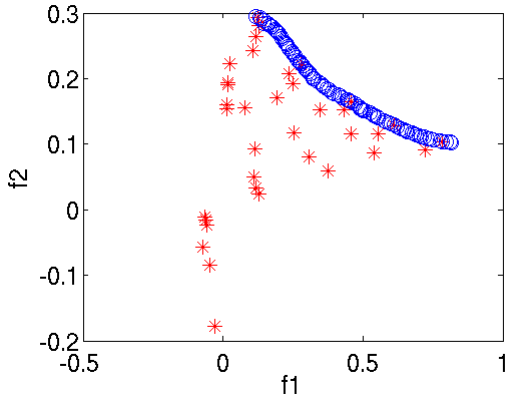
Flapping Wing Design

Find wing parameters (frequency & amplitude) to Maximize Thrust and Efficiency

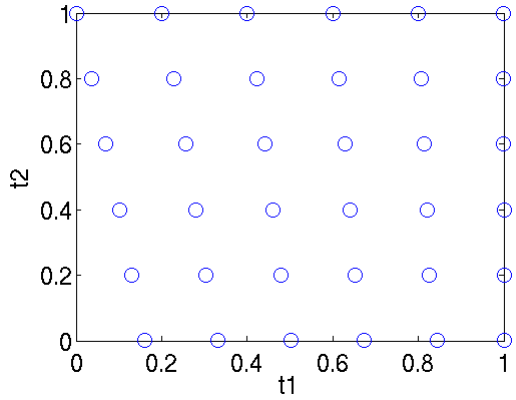


DOE based sampling

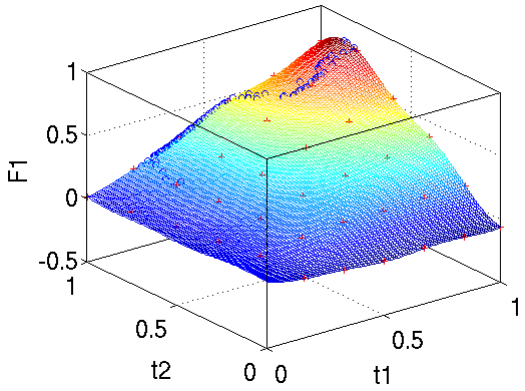


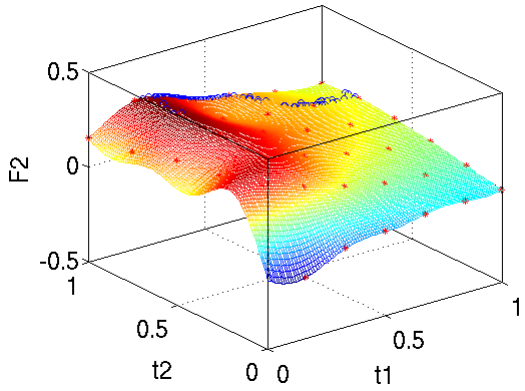


Non-dominated solutions in x-space and f-space



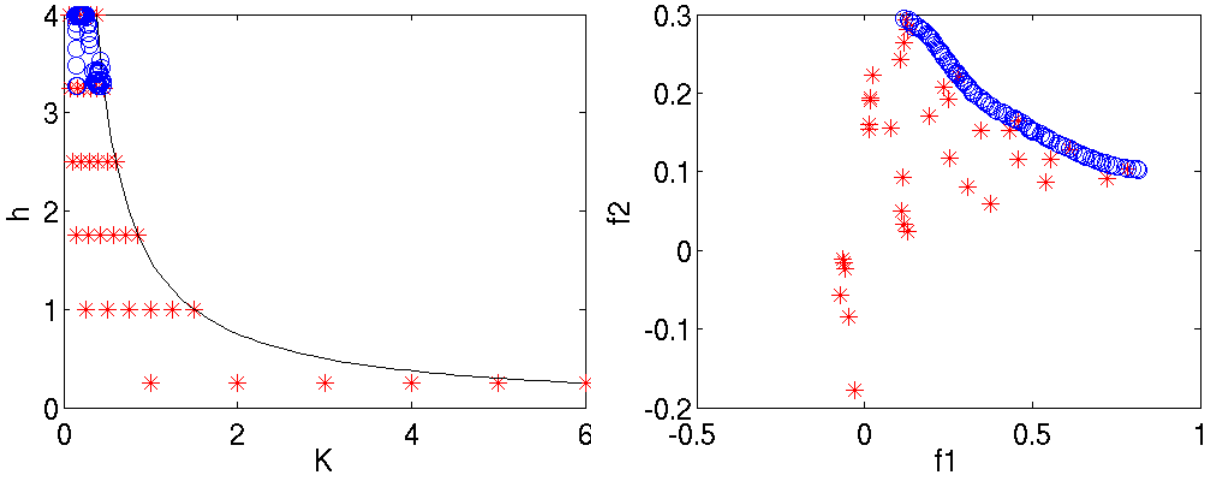
Transformation to parameter space





Surrogate models for f1 and f2 as function of parameters

Flapping Wing Design

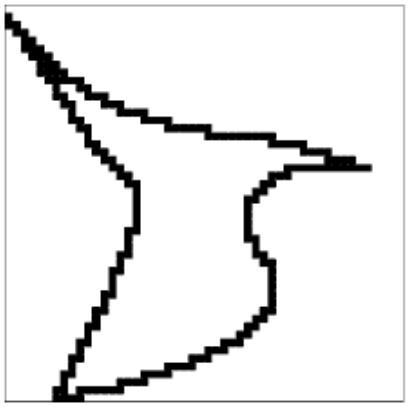


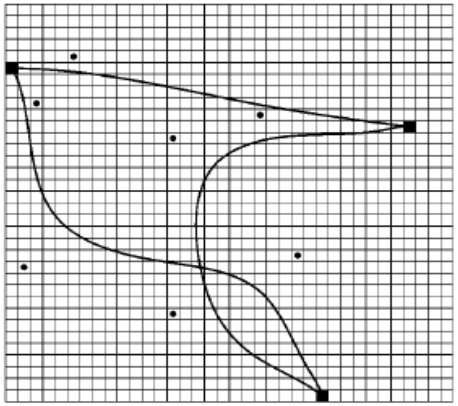
Non-dominated solutions in x-space and f-space

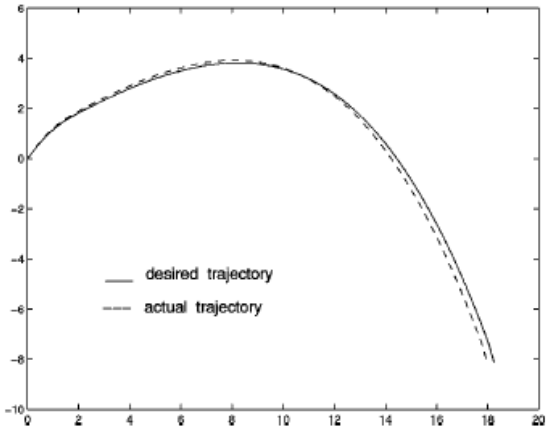
Ashraf, M.A., Isaacs, A., Young, J., Lai, J.C.S. and Ray, T. (2009) Numerical Simulation and Multi-Objective Design of Flow Over Oscillating Airfoils for Power Extraction, *Conference on Modelling Fluid Flow (CMFF'09), Proceedings of 14th International Conference on Fluid Flow Technologies*, Budapest, Hungary, 9-12 September 2009, pp. 221-228..

Parameters		Predicted		Calculated	
Frequency	Amplitude	Thrust	Efficiency	Thrust	Efficiency
3.545	0.4321	0.7896	0.0997	0.7538	0.0972
3.034	0.1403	0.0863	0.305	0.0845	0.287
3.2728	0.1484	0.1176	0.2971	0.1171	0.2963

Topology Optimization of Compliant Mechanisms







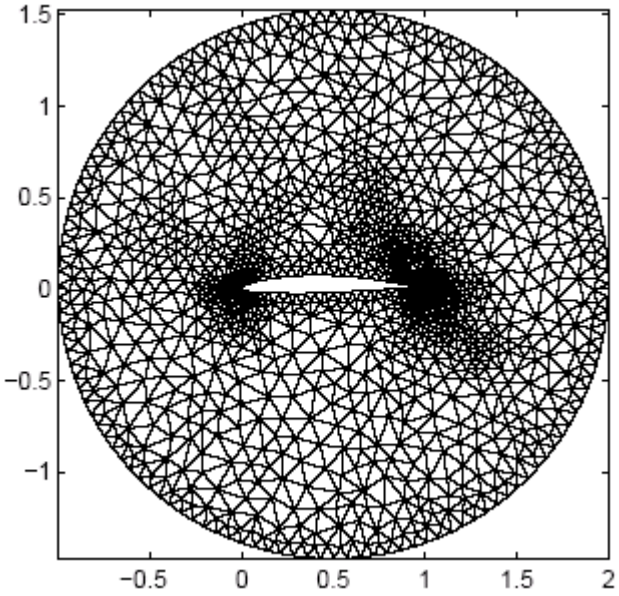
Generate a Topology of the Mechanism such that the tip in the right follows the desired path. EA coupled with ABAQUS.

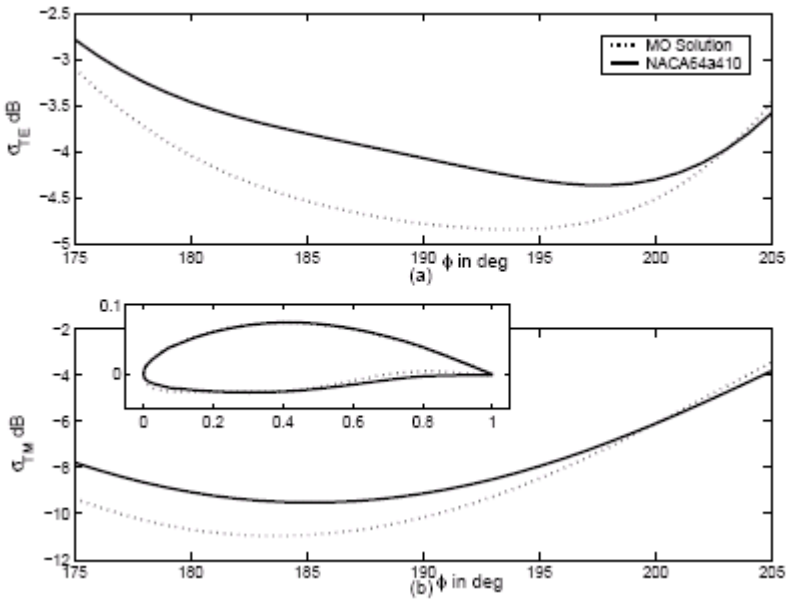
Kang, T., Guang, Y. C. and Ray, T. (2002). Design Synthesis of Path Generating Compliant Mechanisms by Evolutionary Optimization of Topology and Shape, *ASME Transactions, Journal of Mechanical Design*, Vol. 124, September 2002, pp. 492-500.

Airfoil Design with CFD and CEM Considerations

- Minimize F_{CEM}
- Minimize $F_{CFD} = c_d$
- Subject to
  - $c_l = 0.576$
  - $-1^\circ \leq \alpha \leq 1^\circ$

$$F_{CEM} = \max_{\phi \in [180^\circ, 200^\circ]} \{\sigma_{TE}(\phi), \sigma_{TM}(\phi)\}$$

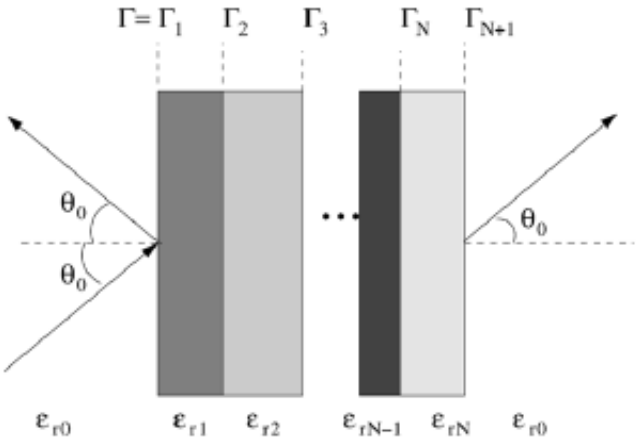


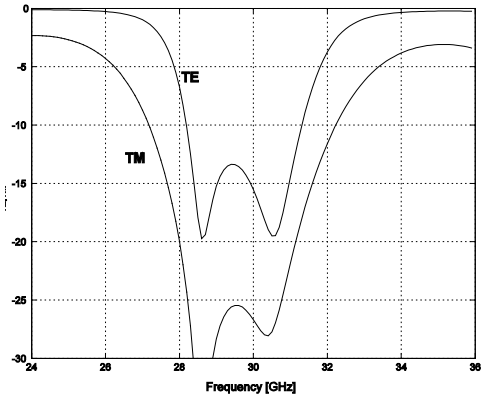


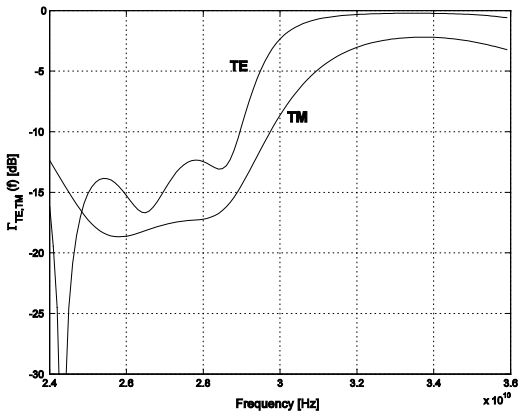
1dB Reduction in Bistatic RCS as compared to NACA64A410

Venkatarayalu, N. and Ray, T. Application of Multi-objective Optimization in Electromagnetic Design, *Real World Multi-objective Systems Engineering: Methodology and Applications*, Eds. Nedjah, N., Nova Science, NY, 2005.

Dielectric Filter Design







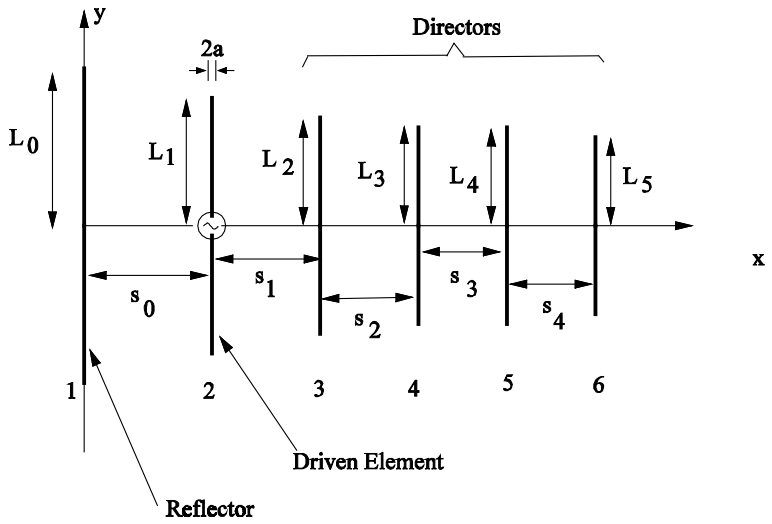
Bandpass Filter Design: Lower cutoff at 28 GHz and Upper cutoff at 32GHZ. Reflection coefficient is greater tha -5dB in stopband and less than -10dB in the passband. & layered dielectric.

Lowpass Filter Design: Cutoff frequency of 30GHZ.

Maximum of 15000 Design Evaluations.

Venkatarayalu, N., Ray, T. and Gan, Y.B., (2005). Multilayer Dielectric Filter Design Using a Multi-objective Evolutionary Algorithm, *IEEE Trans. On Antennas and Propagation*, Vol. 53, No. 11, pp. 3625-3632, 2005.

Yagi-Uda Design



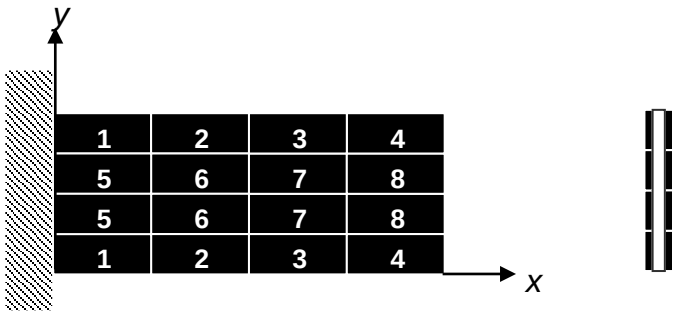
More than 1dBi improvement

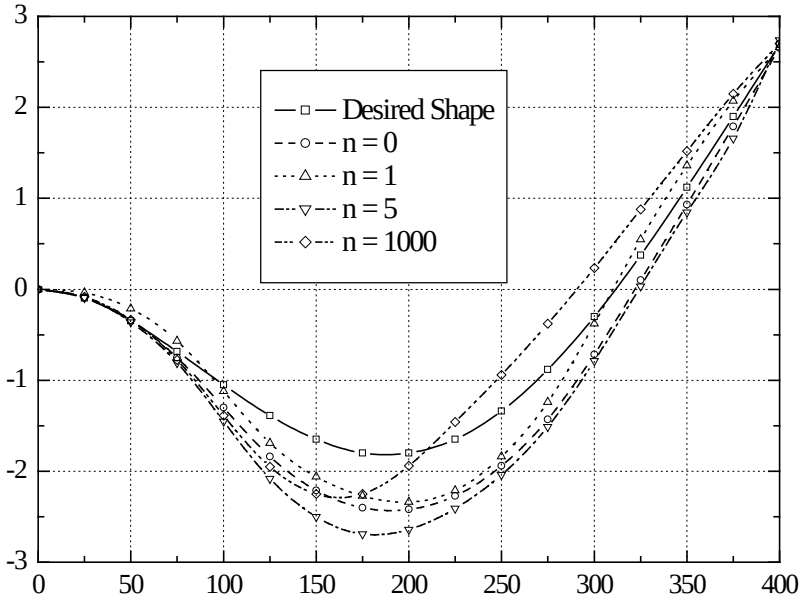
TABLE II
15-ELEMENT YAGI–UDA ANTENNA DESIGNS OBTAINED USING (a) GA FROM [7] AND (b) CI ALGORITHM

	GA optimized [7]		CI optimized	
Elements	Length, L_n	Spacing, S_n	Length, L_n	Spacing, S_n
1	0.236λ	-	0.235λ	-
2	0.230λ	0.249λ	0.227λ	0.196λ
3	0.221λ	0.155λ	0.224λ	0.238λ
4	0.205λ	0.185λ	0.215λ	0.142λ
5	0.216λ	0.191λ	0.204λ	0.231λ
6	0.210λ	0.252λ	0.212λ	0.447λ
7	0.210λ	0.442λ	0.206λ	0.395λ
8	0.189λ	0.431λ	0.203λ	0.371λ
9	0.191λ	0.362λ	0.201λ	0.441λ
10	0.200λ	0.205λ	0.202λ	0.433λ
11	0.204λ	0.268λ	0.206λ	0.445λ
12	0.215λ	0.414λ	0.196λ	0.365λ
13	0.174λ	0.197λ	0.189λ	0.359λ
14	0.199λ	0.130λ	0.203λ	0.429λ
15	0.204λ	0.362λ	0.196λ	0.390λ
Gain(dBi)	15.41		16.66	
Z (Ω)	50..64 – j 5.08		45.42 – j 5.74	

Venkatarayalu, N. and Ray, T. (2004). Optimum Design of Yagi-Uda Antennas Using Computational Intelligence, *IEEE Trans. On Antennas and Propagation*, Vol. 52, No. 7, pp. 1811- 1818, 2004.

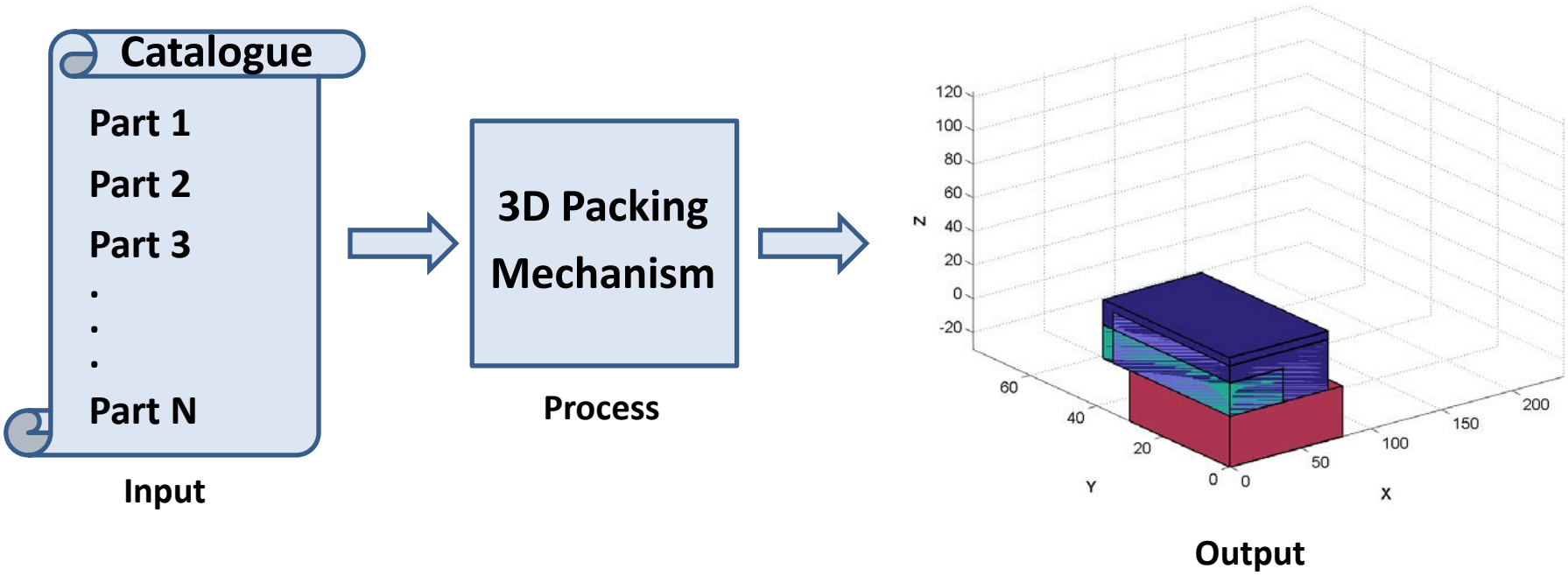
Optimal Gain of Piezoelectric Patches





Liew, K.M., He, X.Q, and Ray, T. (2004). Computational Intelligence in Optimal Shape Control of Functionally Graded Smart Plates, *Computer Methods in Applied Mechanics and Engineering*, Vol. 193, Issues 42-44, pp. 4475-4492, 2004.

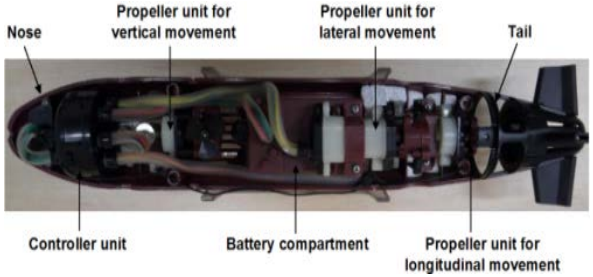
3D Packing Mechanism

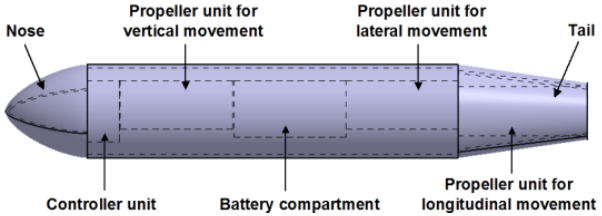


Design Optimization of an Underwater Vehicle

Geometry and Configuration Modules

External Geometry	Internal Components	Designers Requirements
Nose Definition	Power sources	Operating Conditions
Middle Body Definition	Propulsion	Performance Measures
Tail Definition	Payload	Constraints





Analysis Modules

- Drag estimation
- Hydrostatics
- Hydrodynamics
- Control

Optimization Modules

- Non-dominated Sorting Genetic Algorithm (NSGA-II)
- Infeasibility Driven Evolutionary Algorithm (IDEA)
- Surrogate Assisted Evolutionary Algorithm (SA-EA)

Alam, K., Singh, H.K., Isaacs, A., Ray, T. and Sreenatha G. Anavatti (2011), Design Optimization of a Model Submarine: A Reverse Engineering Approach, EUROGEN 2011.

Thank You