

Reversing the design process to aid in complex engineering problems

Amer Farhan Rafique^a, Qasim Zeeshan^b and Ali Kamran^b

^a Mechanical Engineering Department, Mohammad Ali Jinnah University (MAJU), Islamabad, Pakistan

^b Department of Aeronautics and Astronautics, Institute of Space Technology (IST), Islamabad, Pakistan

Email: afrafique@jinnah.edu.pk; afrafique@yahoo.com

Abstract: Design engineers prefer to have multiple performing solutions, rather than a single optimum. This set of design alternatives gives desired flexibility to engineers. The motivation of current research effort stems from the question that “Can the design process be reversed?” Every design endeavor has a specific set of stringent performance requirements to meet. So, if we “reverse” the classical design process and start from those performance requirements, we should obtain a design point meeting such requirements. The traditional inverse engineering methods depends greatly on specific domain knowledge and usually, can effectively facilitate the inversion for a specific application problem. However, it is very difficult to apply them into diverse design scenarios where any of the presumed conditions is violated. Development of a more generic and robust approach is deemed necessary to achieve reliable design results in various design endeavors. This work also aims to answer the question, How to find sufficient design alternatives sufficing a given performance level? In doing so, we also intend to keep check on the computational budget (reduce the total number of function evaluations) even for high dimensional problems.

This work addresses these needs by proposing a methodical approach to identify the feasible region(s), of the large design space of complex problems, containing design points meeting the same or little-less desired performance level. In this way, continuous and/ or discontinuous segments of design space can also be identified. Such regions are anticipated to meet acceptable performance levels. The proposed approach can be cast as a rough set based design methodology. The procedure identifies the design spaces corresponding to the required objective function value, by extracting rules from input-output information system, instead of an approximation of the objective function. The discretized decision system/ table and extracted rules act as transparent metamodel establishing relationship between performance space and design variable space. Thus the proposed method can identify multiple global optima in contrast to a single optimum identified by traditional global optimizers. Latinized Hypercube Sampling is employed to generate information/ decision system to identify attractive spaces even for complex high dimensional problems, thus, limiting total function evaluations to a modest number.

The performance of the proposed procedure has been tested and, thus, validated by the trajectory modelling problem. The inverse design approach based on rough sets is intended for initial conceptual design purposes, thus, providing an immediate insight on the performance prior to the detailed design phase.

Keywords: *Engineering design, reverse design process, optimization, trajectory modelling, air launched satellite launch vehicle.*

1. INTRODUCTION

Modelling the engineering design problem as an optimization problem in search of global optima has been the trend in most of the approaches used to locate optimum design. Optimization of engineering design problems through stochastic approaches are being extensively probed in the last few decades (Rafique *et al.*, 2011). The basic scheme of these methods is to solve and modify trial settings of design variables through an iterative procedure. Computational cost is one of the major drawbacks of these approaches, especially when the design problem under consideration is complex involving numerous competing disciplines. In recent times, unconventional procedures based on artificial neural networks, response surface and support vector machines have been proposed (e.g. Qazi and Linshu, 2006; Papila and Wei, 2001). A two phase neural network has been proposed for forward and reverse modeling of a manufacturing process, which is designed for determining cutting parameters in wire-EDM (Electrical Discharge Machining) (Wang, 2004).

In particular, in complex engineering design endeavors, these techniques approximate the objective function by employing optimization algorithms on a metamodel of the objective instead of on an objective function itself. These procedures, coupled with optimization algorithms, greatly reduce the computational cost by performing the analysis on the metamodel. These approaches have been successfully used for engineering conceptual design systems, but these methods are time consuming, cumbersome and are iterative in nature. One major challenge is the approximation quality of a metamodel and the tedious training requirement for artificial intelligence tools. This work also intends to keep check on the computational budget (reduce the total number of function evaluations) even for high dimensional problems.

Design engineers prefer to have multiple performing solutions, rather than a single optimum. This set of design alternatives gives desired flexibility to engineers. The motivation of the current research effort stems from the question that “Can the design process be reversed?” Every design endeavor has a specific set of stringent performance requirements to meet. So, what if we “reverse” the classical design process and start from those performance requirements to reach at the design point meeting such stringent performance requirements? The traditional inverse engineering methods depend greatly on specific domain knowledge and usually, can effectively facilitate the inversion for a specific application problem. However, it is very difficult to apply them into diverse design scenarios where any of the presumed conditions is violated. Development of a more generic and robust approach is deemed necessary to achieve reliable design results in various design endeavors. The intrinsic multidisciplinary nature of conceptual design of engineering systems presents significant and exciting research challenge, and thus, potential candidates for the proposed inverse design approach.

This work addresses these needs by proposing a methodical approach to identify the feasible region(s), of the large design space of complex problems, containing design points meeting the same or little-less desired performance level. Such regions are anticipated to meet acceptable performance levels. The procedure identifies the design spaces corresponding to the required objective function value, by extracting rules from input-output information system, instead of the approximation of the objective function. The performance of the proposed procedure has been tested on the trajectory modelling problem and compared by the previously attempted study of the same optimization problem (Rafique *et al.*, 2010). The optimal design space of the trajectory modelling problem is modeled as an inverse design problem and feasible sets are found satisfying performance criteria and constraints.

Followed by introduction, three sections cover this paper. The first section covers proposed inverse design approach and theory of the rough sets. The second section compares the engineering design problem of trajectory modelling by proposed approach and also presents results. The last section consists of brief discussion and conclusion followed by future work.

2. INVERSE DESIGN APPROACH

Under current consideration design alternatives are termed as Design Variable Space (DVS). The attributes to distinguish alternate designs in DVS are termed as design variables a_1, a_2, \dots, a_m . The next step is to obtain the Performance Variable Space (PVS) through a mapping function f . Aspects of performance of design are termed as performance variables d_1, d_2, \dots, d_q . Each d_j is mapped by f_j such that $d_j = f_j(a)$. PVS is, thus, the set of quantified performance variables achievable within DVS. For a certain instance of design problem, only a part of DVS and PVS will be involved which are called “Valid Design Variable Values” and Valid Performance Variable Values”, respectively.

This approach proposes the rough set based decision system to capture and interpret the functional relationship between the DVS and the PVS. The purpose of proposed procedure is to determine the inputs that correspond to a prefixed output.

Rough-sets were first applied successfully for small test problems in the mechanical design area by Shan and Wang (2004). This method has limitations with high dimensionality engineering design problems. The proposed approach in this research effort aims to remove this limitation. The following section presents the brief theoretical background of the theory.

2.1. Rough Set Theory

In 1982, a new theory of rough sets to construct estimation of concepts from obtain data was presented by Pawlak (1982). Over the years, this theory has seen significant developments. Rough sets have been proven to have ability to tackle practical engineering problems, thus attracting more and more researchers from various domains.

Easy understandability, non-requirement of preliminary information about data, efficiently reduction of original data to match performance parameters, and applicability for parallel processing are few of the many fundamental advantages of rough sets theory.

Working of rough sets can be summarized as under;

- Information system (S) is initially created which is a function of two parameters. These parameters are finite and non-empty samples of the data, called universe (U) and finite set comprising of attributes (A). This finite set of attributes will map the data samples of universe into value sets (V_a).
- The target system is classified as a decision system (d) represented by a decision table of finite values of conditional and decision attributes for all S .
- The decision system is then ranked by $r(d)$ to create partitions of attributes (P_a) in the universe.
- All values of P_a are defined by the set of cuts (C_a) on V_a . This C_a will then define a new conditional attribute and the whole system is repeated until all attributes are mapped.

2.2. Proposed Inverse Design Approach

The significances of the proposed inverse design approach are twofold. First, it offers an instinctive method to identify the design spaces corresponding to the required objective function value (performance). Secondly, it gives designers the flexibility and option to select the solution from several equally good solutions.

Figure 1 illustrates the proposed inverse design approach and described as follows;

- **Step 1 – Design-of-Experiments:** Perform the sampling of entire design space and get the corresponding objective function value using exact multi-disciplinary analysis. Latinized Hypercube Sampling (LHS) is used as the sampling strategy for information system because of its efficient and evenly space filling properties. This property aids in reducing the likelihood of missing vital features of design space.
- **Step 2 – Choosing a decision threshold d_t :** Decision threshold should lie between a minimum and a maximum function value and is a real number. Normally, a slightly greater value than extreme minimum is selected. It is actually a bracketing for objective function values.
- **Step 3 – Construct decision system S :** Form a table in which each row represents one sample (from step 1) and the last column lists the corresponding classification value (0 or 1) of objective function based on the decision threshold selected in step 2. $D = 1$ is assigned if the sample has function value greater than d_t , otherwise $D = 0$.
- **Step 4 – Propositional variable P_a :** This step involves the introduction of a propositional variable that corresponds to the respective attribute in the decision system.
- **Step 5 – Interval cuts C_a :** Interval cuts are simply the mean of each propositional interval (each P_a corresponds to C_a within its interval).
- **Step 6 – Construct S^* from S :** Only those pairs of objects from U are listed as S^* who have different decision values.

- **Step 7 – Design space cuts:** A column from step 6 (S^*) with maximum occurrences of 1 is chosen herein. If the number of such columns is more than one then random selection from any one of them is made. Replace all 1's with 0's in the selected column.
- **Step 8 – Loop and list of cuts:** Step 7 is performed in loop till all entries in S^* become zero. Create a list of all chosen columns or propositional variables and their corresponding cuts. These sets of cuts divide the original design space into several smaller spaces of attributes having the same decision value.

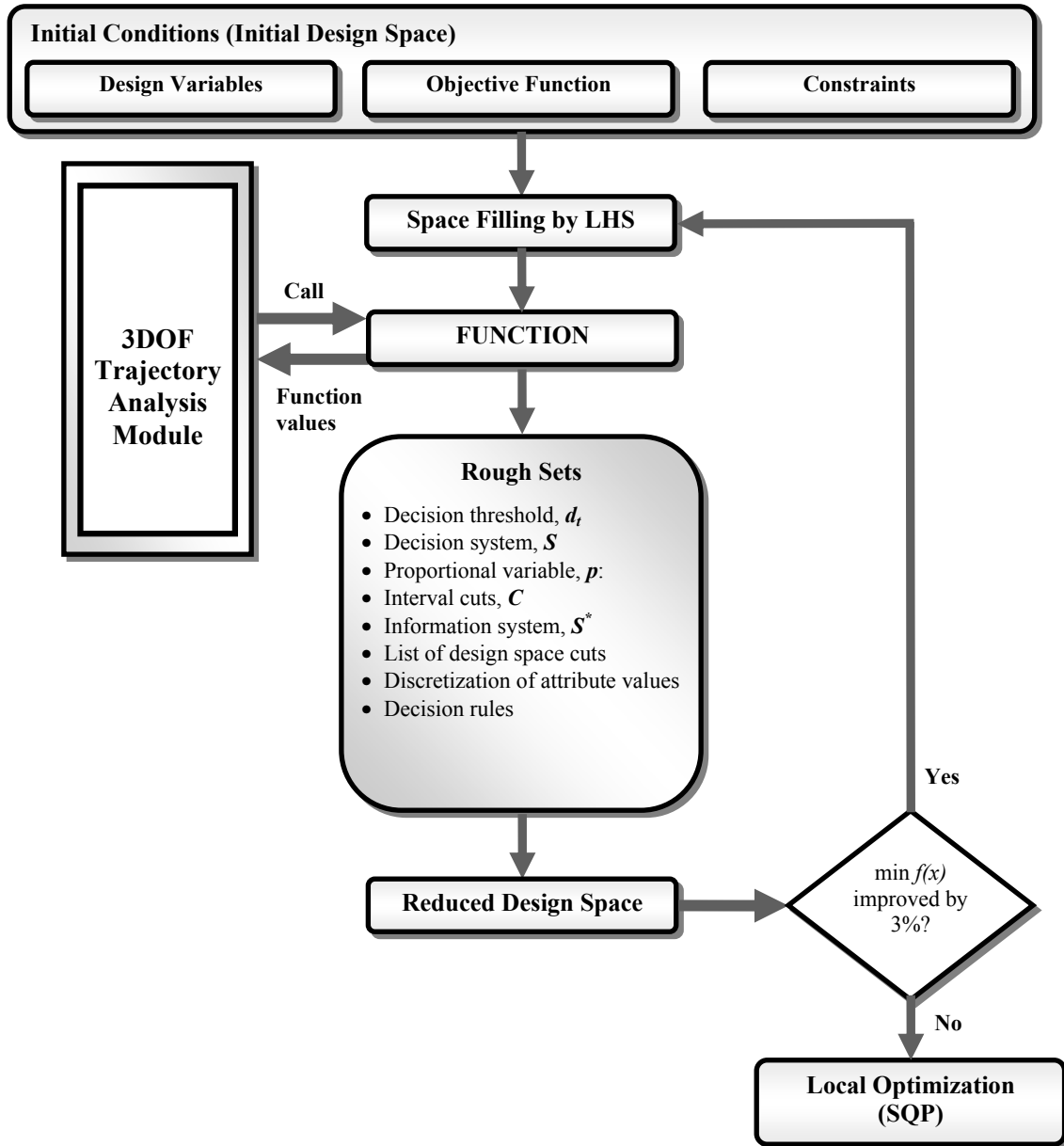


Figure 1. Proposed Inverse Design Approach

- **Step 9 – Discretization of attribute values:** Step 7 and 8 will result in attribute intervals each having integer values. Decision rules are generated from these integer values. This operation reduces the size of values attribute sets. This process is then repeated for all variables.
- **Step 10 – Generation of decision rules:** The purpose of this step is to eliminate the surplus information created from the preceding steps to reach at the table containing decision rule for each variable with target attribute value.
- **Step 11 – Identification of attractive design space and new bounds:** This step is the backbone of the proposed approach based on rough sets. Attractive regions of design space that approximate the desired

performance level are identified in this step. The user specific decision threshold of objective function and the decision value corresponds to the desired function performance extracts the attractive design space for the specific objective function. New lower and upper bounds define these attractive regions and these small regions will be used for sampling and searching the optimum in the next iteration.

- **Step 12 – Iteration loop and termination criteria:** This step takes the iterative loop to step 1 and repeats the whole process with sampling only in the identified attractive regions. Find $Min f(x)_{i+1}$ for current iteration and compare with $Min f(x)_i$. If the improvement in current minimum of objective function and previous iteration is more than 3% then continue, else Stop.
- **Step 13 – Local optimization:** Perform local optimization using Sequential Quadratic Programming (SQP) in all the identified attractive regions to get multiple global optima.

3. TRAJECTORY MODELLING PROBLEM (COMPARISON)

The proposed rough sets based inverse design approach has been compared against the global optimization problem of trajectory modelling of an air launched SLV (Rafique *et al.*, 2010) for the purpose of validation of the proposed inverse design approach. Efficient trajectory modelling and optimization is a basic ingredient for successful mission completion. An optimum ascent pattern leads to efficient performance of the vehicle. Air launched SLV undergoes a launch maneuver through precise AOA profile after few seconds of horizontal launch. Subject AOA profile is imperative to successful completion of a specific mission scenario. The trajectory problem is modelled to attain maximum altitude (objective function). Axial overload, lateral overload and maximum allowable AOA are considered as constraints of this problem. Launch maneuver variable (a_m), time to start power ascent phase (t_1), and bounded maximum AOA (α_{max}) are the design variables for this problem. Outputs from this problem are then inputted to 3-DOF trajectory model simulated in MATLAB. Figure 2 presents the ascent phase of said problem. Governing equations of this problem are as follows;

$$\alpha_{prog}(t) = a_m \sin^2 f(t) \quad (1)$$

$$f(t) = \frac{\pi(t-t_1)}{k(t_2-t) + (t-t_1)} \quad (2)$$

$$k = \frac{t_m - t_1}{t_2 - t_m} \quad (3)$$

$$n_x = \frac{T + L \sin \alpha - D \cos \alpha}{mg_o} \leq n_{xmax} \quad (4)$$

$$n_y = \frac{T \sin \alpha + L}{mg_o} \leq n_{ymax} \quad (5)$$

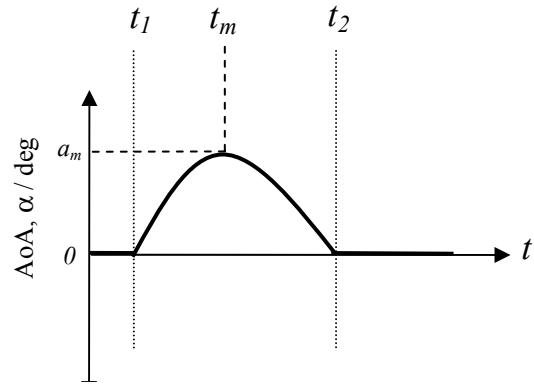


Figure 2. AOA profile

Where, t is time of flight, t_2 is time at the end of turning phase, t_m relates time corresponding to maximum AOA, n_x and n_y are axial and lateral overloads respectively.

The objective of the problem is to maximise peak altitude reached whilst adhering to the constraints (Eq. 6). There were three design variables for this problem as presented in Eq. 7. This problem is a mix of inequality and equality constraints, total of 5 inequality and 2 equality constraints (Eq. 8-9). The optimization process makes use of dynamic penalty function to handles the whole optimization problem by adding constraints in the objective function. Equation 10 represents the symbolic problem statement.

$$f(x) = \max alt_f(x) \quad (6)$$

$$x = \begin{bmatrix} a \\ \alpha_m \\ t_1 \end{bmatrix} \quad (7)$$

$$C_i \geq 0, \quad i = 1,2,\dots,5 \quad (8)$$

$$C_i = 0, \quad i = 6,7$$

$$C1 = n_{x1} \leq 11$$

$$\begin{aligned}
 C1 &= n_{x1} \leq 11 \\
 C2 &= n_{x2} \leq 11 \\
 C3 &= n_{y1} \leq 2 \\
 C4 &= q \leq q_{\max} \quad (\text{at } \alpha = 0) \\
 C5 &= q \leq q_{\max} \quad (\text{at separation of 1}^{st} \text{ and 2}^{nd} \text{ stage}) \\
 C6 &= \alpha = 0 \quad (0.8 \leq M_a \leq 1.3) \\
 C7 &= \vartheta = 0 \quad (\text{at orbit insertion})
 \end{aligned}
 \tag{9}$$

$$\max alt_f(x) = f(x) + h(k) \sum_{i=1}^m \max\{0, g_i(x)\}
 \tag{10}$$

Where, $f(x)$ is the objective function, $h(k)$ is modified penalty value and k is the current iteration number. The function $g_i(x)$ is a relative violated function of the constraints (Crossley and Williams, 1997).

The results show successful implementation of inverse design approach and identification of multiple attractive sub-spaces. Our method only required a total of 4180 function evaluations to find five such solutions (Table 1). As shown, all five solutions retrieved by the inverse design approach were globally or near-globally optimum. However, previous attempt required 10000 function evaluations to reach an optimal point (Rafique *et al.*, 2010). Also, this traditional method only found one global optimum in each run whereas, our proposed method found five solutions in a single run including global optima. Furthermore, these 4180 function evaluations by the inverse design approach are enough to identify the feasible design spaces and then to perform further local optimization in these spaces.

Table 1. Comparison of function evaluations

Proposed Inverse Design Approach + SQP					GA			
Run	Fn. Evaluations in Rough Set iterations	SQP Fn. Evaluations	Total Fn. Evaluations	Optimal Solution (Alt-km)	Run	Fn. Evaluations in GA	Total Fn. Evaluations	Optimal Solution (Alt-km)
1	3221	198	4180	510	1	100000	100000	500
		174		501	2	100000	100000	496
		202		502	3	100000	100000	504
		190		508	4	100000	100000	501
		195		507	5	100000	100000	490

Table 2 shows the iterations of rough set based inverse design approach. The number of attractive regions identified during iterations is presented in second column. 3rd ~ 6th column lists, respectively, the number of samples evaluated in each attractive region, total function evaluations performed to identify the attractive regions in each iteration, the decision threshold d_i and the number of samples that fall below the decision threshold. The last column of this table shows the value of best objective function found in each iteration. Table 3 presents the optimal design points from inverse design approach and compared against the previous study. It can be seen that though slight improvement in objective function value is achieved through inverse design approach but major contribution of this approach is reduction in number of function evaluations to a great extent. Constraints of axial overloads, lateral overloads and max AOA are not violated.

Table 2. Iterations of inverse design approach

Iteration	No. of attractive regions	No. of samples in each region	Total Fn. Evaluations	Decision threshold selected	Samples below threshold	Max. Objective Fn
1	1	509	509	57	10	475
2	6	214	1284	42	6	491
3	4	357	1428			491

The size of information system S^* (step 6) can grow extremely large if the sampling strategy is not selected carefully. The number of columns depends on the number of unique partitions for every design variable. For random sampling size of say 500, there will be around 500 unique partitions for each variable. In the proposed method, with 509 samples of LHS there were only 102 unique partitions (equal sized) for each variable, so there were only $03 \times 128 = 306$ columns in S^* . The number of rows of S^* depends on how many

samples fall below the decision threshold, so the value of decision threshold is critical. In first iteration, there were 10 out of 509 samples fall below the threshold.

Table 3. Optimal solution comparison

Design Variable	Symbol	Unit	LB	UB	Optimal Solution GA	Optimal Solution Inverse Design Approach
Launch Maneuver Variable	a_m		0.01	0.20	0.0146	0.0134
Time of Start of Power Ascent Phase	t_i	s	1.00	5.00	1.89	1.47
Maximum Angle of Attack	α_{max}	deg	1.00	22.0	21.63	21.51
Objective Function (Maximum Altitude)					501 km	510 km

4. CONCLUSION AND DISCUSSION

This present work has contributed the field in following two ways;

- It is the first instance of the inverse design approach being successfully applied to an aerospace problem.
- Latinized Hypercube Sampling and its integration with inverse design approach using rough set tools enables in identifying multiple attractive regions and optima with a limited number of function evaluations for a high dimensional problem.

An algorithm able to solve inverse synthesis engineering design optimization problem based on the rough set approach has been presented. The set of rules extracted by rough set approach capture the functional relationship between design parameters and specific chosen design targets. The proposed method successfully identifies attractive spaces based on the chosen decision thresholds, with significantly reduced function evaluations. The proposed methodology has taken a different route in solving complex engineering problems and yielded ‘small islands’ in a fairly bigger design space. It is expected that that all solutions in such islands will have objective function value less than provided threshold.

The performance of the algorithm has been tested against the trajectory modelling problem of air launched SLV and compared with optimization using Genetic Algorithms. Results show improved performance in terms of computation time, due to the lesser number of exact analyses. The proposed method is capable of identifying multiple global optima in contrast to single optimum found by traditional global optimizers. Moreover, the use of an inverse design approach is principally appropriate for tackling the design scenarios engineering system problems where the objective function is extremely heavy to be evaluated, as for example when it has to be defined through numerical methods like, CFD and FEM.

REFERENCES

- Rafique, A.F., LinShu, H., Kamran, A., Zeeshan, Q. (2011). Hyper heuristic approach for design and optimization of satellite launch vehicle. *Chinese Journal of Aeronautics*, 24(2), 150–163.
- Papila, N. and Wei, S. (2001). Shape optimization of supersonic turbines using response surface and neural network methods. 39th Aerospace Sciences Meeting and Exhibit, Reno, NV, 8 – 11 January.
- Qazi, M. and Linshu, H. (2006). Efficient sampling and support vector regression for multidisciplinary design optimization of multistage space launch vehicle. 44th AIAA Aerospace Sciences Meeting, Reno, 9 – 12 January.
- Wang, G.N. (2004). Two-phase reverse neural network approach for modeling a complicate manufacturing process with small sample size. *Neural Information Processing – Letters and Reviews*, 2(1).
- Rafique, A.F., LinShu, H., Kiani, M.U. and Ali, S.A. (2010). Trajectory optimization of air launched satellite launch vehicle using genetic algorithm. ICNPAA 2010 World Congress: 8th International Conference on Mathematical Problems in Engineering, Aerospace and Sciences, 611–619.
- Shan, S. and Wang, G. (2004). Space exploration and global optimization for computationally intensive design problems: A rough set based approach. *Journal of Structural and Multidisciplinary Optimization*, 28(6), 427–441.
- Pawlak, Z. (1982). Rough sets. *International Journal of Parallel Programming*, 11(5), 341–356.
- Crossley, W.A. and Williams, E.A. (1997). A study of adaptive penalty functions for constrained genetic algorithm-based optimization. 35th AIAA Aerospace Sciences Meeting and Exhibit, Reno, 6 – 9 January.