A Large-scale Bi-objective Optimization of Solid Rocket Motors Using *Innovization*

Abhiroop Ghosh, Erik Goodman, Kalyanmoy Deb
Department of Electrical and Computer Engineering
Michigan State University
East Lansing, USA
{ghoshab1, goodman, kdeb}@egr.msu.edu

Ronald Averill, and Alejandro Diaz
Department of Mechanical Engineering
Michigan State University
East Lansing, USA
{averillr, diaz}@egr.msu.edu

***Abstract*—Many design optimization problems from practice involve a large number of variables. In handling such problems, optimization algorithms, in general, suffer from the well-known "curse of dimensionality" issue. One of the ways to alleviate the issue somewhat is to use problem information to update the optimization algorithm so that more meaningful solutions are evolved quickly. In this paper, we consider a solid rocket motor design problem involving hundreds of integer variables and two conflicting objectives – minimize the error in matching developed thrust with a desired time-dependent thrust profile and simultaneously minimize the unburnt residue of propellant at the end of the burning process. The evaluation of both objectives involve a detailed burn simulation from the core to the shell of the rocket. After finding a set of trade-off solutions using an evolutionary multi-objective optimization algorithm, we use two learning-based optimization methods (akin to the concept of *innovization*) to find similar set of solutions using a fraction of the overall solution evaluations. The proposed methods are applied to seven different thrust profiles. Besides solving the large-scale problem quicker, a by-product of our approach is that learnt innovized principles stay as new and innovative knowledge for solving the solid rocket design problem, a matter which is extremely useful to the practitioners.**

***Index Terms*—Solid rocket motor design, Multi-objective optimization, *innovization*, multi-objective optimization.**

I. Introduction

Multi-Objective Optimization (MOO) algorithms are widely used in many design problems. They produce a set of solutions representing a trade-off between two or more objectives known as Pareto-optimal (PO) solutions. The decision-maker can then select a solution from the set. Among the PO solutions there might exist hidden relations or common characteristics which can give some insight into the design problem. The process of determining these characteristics is referred to as *innovization*, first proposed by Deb and Srinivasan [1].

A clustering-based technique is suggested in [2] for automation of the innovization process. The concepts of higher- and lower-level innovizations are presented in [3]. An automated innovization technique for handling discrete variable problems was proposed in [4].

Recent works have focused on introducing the results of an innovization study, or in other words, the innovized *rules* or *principles*, back into the optimization process as a heuristic. This has been demonstrated in [5] on a machining parameter optimization problem where innovized principles were used in a local search; faster convergence achieved as a result. An interleaving of data mining and MOO methods was used in [6] to achieve faster convergence. Gaur and Deb [7] proposed an adaptive innovization method which repairs the solutions directly based on the innovized rules learned using machine learning techniques.

This study introduces a solid rocket design problem and formulates it as a multi-objective optimization problem. The complexity in this problem mainly arises from the large number of decision variables and with all of them being discrete. Initially, the problem is solved using a specific evolutionary multi-objective optimization (EMO) method – NSGA-II [8]. Subsequently, two innovization approaches are presented with the aim of speeding up convergence. The performances of all three methods are presented in this paper.

The paper is organized as follows. Section II describes the rocket design problem. Section III describes the optimization problem formulation. Section IV describes the NSGA-II algorithm used for the rocket design problem and the corresponding results. Two specific innovization approaches are introduced in Section V and finally, the conclusions of this extensive study are presented in Section VI.

II. Rocket Grain Design Problem Formulation

The objective of this problem is to model a solid fuel rocket motor and find an efficient method to determine the best possible design for the grain (propellant distribution) which will provide the specified burnout characteristics (thrust versus time). A second objective is to minimize or eliminate insulation by ensuring the propellant burns out simultaneously throughout the rocket. Each kilogram of insulation thus eliminated yields an additional kilogram of payload.

The target thrust profiles are shown in Fig. 1. The proposed methods are first tested on the baseline thrust profile, and then extended to the other thrust profiles.

The following are the objectives and constraints:

- The rocket should burn for at least as long as the target thrust profile is defined, which for the baseline thrust profile, is 10 seconds.
- The target thrust profile should be matched within 5% at every point in time.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

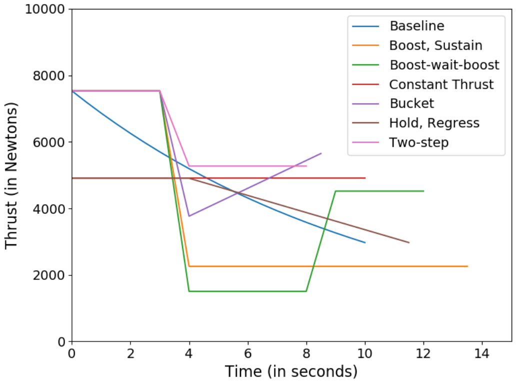


Fig. 1: Target thrust profiles.

- For the primary burn portion, the pressure should lie within a certain range. For the baseline thrust profile, the desired pressure range is 1.379 MPa to 3.447 MPa.
- The specified insulation is required at the nozzle in order for the rocket burn properly.

The assumptions are as follows:

- Propellant density throughout the rocket is assumed to be constant, despite changes in reference burn rate.
- Ignition transients are ignored and all exposed propellant surfaces are assumed to start burning at the same time.
- Rate of mass flow into the combustion chamber is considered equal to the rate of mass flow out of the rocket.

11 propellant types are available at each location, numbered from 0 to 10 in increasing order of burn rates, as presented in Table I.

TABLE I: Available propellant types and their burn rates.

Type	Reference Burn Rate (m/s)
0	0.00254
1	0.00305
2	0.00363
3	0.00434
4	0.00521
5	0.00622
6	0.00744
7	0.00892
8	0.01064
9	0.01275
10	0.01524

For this study, a rocket burn simulator developed at Michigan State University by the second author has been used. Given a design, it simulates the burn throughout the rocket. Simulation terminates if one of the following conditions occurs:

- The burn hits the shell at any point.
- The pressure goes outside the allowable range.

The rocket model is illustrated in Fig. 2. The rocket has been vertically divided into segments numbered from 0 to 12 with each segment divided horizontally into 20 layers. Propellants are specified for each layer of every segment. Segment 0 represents the dome. Segments 1 to 5 have the propellants arranged in the form of concentric cylindrical ”rings”. In addition, segments 2 to 4 (here called ”star” segments) have some additional propellant arranged in a non-cylindrical fashion (the finocyl cross-sections shown in Fig. 3). This provides the additional initial surface area necessary to enable the rocket to achieve a variety of thrust profiles. Segments 6 to 11 are called corner segments, which exist between the dome and the cylindrical segments, and each of which has a different length. Segment 12 represents the nozzle section. Fig. 2 shows a rocket having 9 layers per segment and the side-view of the burn at 0, 3.5 and 9 seconds after ignition. Fig. 4 shows the top-down view of the cylindrical and star segments during the burn.

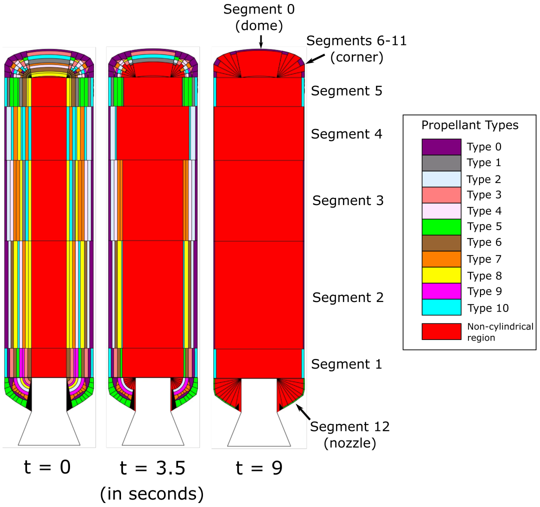


Fig. 2: Side view of rocket during burn. Red region shows the space which is already burnt or was originally empty.

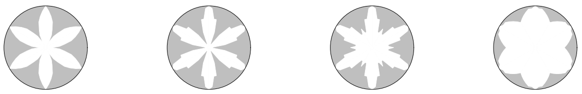


Fig. 3: Examples of different star-like (finocyl) shapes that can be generated by the rocket model. Propellants are represented in gray.

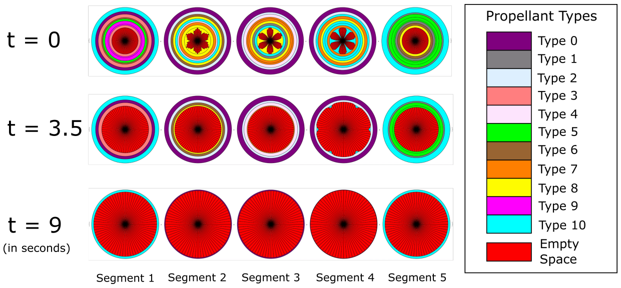


Fig. 4: Top view of rocket during burn on five segments.

III. Optimization Problem Formulation

A multi-objective formulation of the solid rocket design problem has been developed which attempts to optimize two conflicting objectives – (i) minimum thrust profile error, and (ii) minimum uniform burnout residue, simultaneously:

$$\begin{aligned}
\text{Minimize} \quad & f_1 = \sum_{t=0}^{T_{burn}} (F_T(t) - F_o(t))^2, && (1)\\
& f_2 = \mu_r + \sigma_r, && (2)\\
\text{subject to} \quad & P_{min} <= P(t) <= P_{max}, && (3)\\
\text{where} \quad & T_{burn} \equiv \text{Target burn time},\\
& F_T(t) \equiv \text{Target thrust at time } t,\\
& F_o(t) \equiv \text{Thrust obtained at time } t,\\
& \mu_r \equiv \text{Mean of segment residues},\\
& \sigma_r \equiv \text{Standard deviation of segment residues},\\
& P(t) \equiv \text{Pressure at time } t,\\
& P_{min} \equiv \text{Minimum allowable pressure},\\
& P_{max} \equiv \text{Maximum allowable pressure}.
\end{aligned}$$

Equation (1) represents the sum of squared errors between the target $F_T(t)$ and obtained thrusts $F_o(t)$ at all points in time. If $F_o(t)$ is within 5% of $F_T(t)$ at time t, then the error at that time is considered to be zero.

Equation (2) represents the simultaneous burnout objective. A well-designed rocket should burn for as long as possible. μ_r is included in the objective to encourage the optimizer to reduce the amount of propellant left at the end of burn. For the rocket to achieve almost simultaneous burnout, the residue left in all the rocket segments at the end of the burn should roughly be the same. Hence σ_r needs to be minimized as well to ensure uniformity in residue across different segments.

Equation (3) represents the pressure constraint which ensures the pressure at any point during the burn stays within allowable levels. A good match with the target thrust automatically ensures the satisfaction of this constraint.

There are a total of 284 inputs to the rocket burn simulator, as shown in Fig. 5. Among them, 227 are the decision variables for the optimizer to determine. The various types of decision variables are represented in Table II.

TABLE II: Decision variable properties.

Variable	Type	Range	Number
Cylindrical layer propellants	Discrete	$[0, 10]$	203
Star ray depth codes	Discrete	$[0, 3]$	18
Star section propellants	Discrete	$[0, 35]$	3
Propellant for circularization	Discrete	$[0, 10]$	3

In this study, thickness of the individual layers is kept constant to reduce the number of decision variables in the optimization problem. Since the burn starts from the innermost layers and only moves to the outer layers after some time, there are both spatial and temporal aspects to this problem. Certain decision variables are never expressed if a particular rocket design does not burn beyond a particular time or if the burn does not continue long enough to reach the shell.

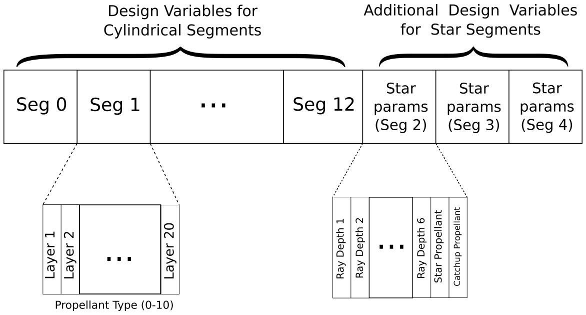


Fig. 5: Rocket input parameter vector.

IV. Optimization Using NSGA-II

NSGA-II [8] has been used to solve the optimization problem presented in Section III. It is an elitist non-dominated-sorting-based Genetic Algorithm (GA). In every generation, an offspring population is generated from the parent population using mutation and crossover. Non-dominated sorting is performed on the combined parent and offspring population and different non-dominated fronts are identified. The combined population ensures elite solutions are preserved across generations. NSGA-II uses a crowded tournament selection operator which initially selects the solutions according to the non-dominated fronts they belong to. A tie between two solutions lying on the same non-dominated front is resolved by choosing the one having a greater crowding distance. This acts as a diversity-preserving mechanism.

A. Results Using NSGA-II

This section presents the results obtained on the optimization problem defined in Section III. Seven thrust profiles used in this study are shown in Fig. 1. The original NSGA-II [8] algorithm is used on these problems without any modification. This optimization is referred to as "Base NSGA-II optimization" for the rest of the paper. The NSGA-II implementation in the *pymoo* framework [9] was used. A population size of 500 was used and the maximum number of function evaluations set to be 40 million. The final population and the Pareto-optimal Front for the baseline thrust profile are shown in Fig. 6.

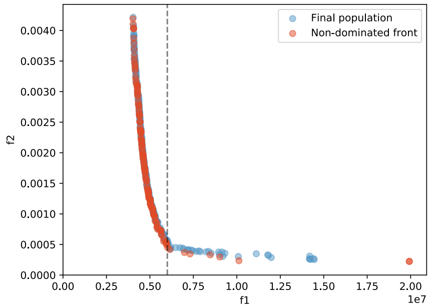


Fig. 6: Pareto-optimal front for the baseline thrust profile.

B. Discussions

In this problem, the Pareto-optimal Fronts obtained for all the target thrust profiles show a *knee* region [10]–[14]. For the baseline thrust profile, the knee region is where the black dashed line passes through the Pareto-optimal Front in Fig. 6. The solutions lying to the left of the knee region show a significant improvement in thrust-match objective while incurring a small deterioration in the residue objective. The opposite is true for the solutions lying to the right of the knee region. Since the primary objective is to minimize the thrust-match, we concentrate on the solutions lying to the left of the knee region. The thrust profile and residues across all the segments for a solution close to the knee are shown in Fig. 9a. From the figure, it is evident that NSGA-II has mostly been able to match the target thrust profile until 10 seconds. The rocket is also able to burn for some time beyond 10 seconds. And the residue for each segment is in the order of 1 mm or less. The simulation terminated because the burn in Segment 1 hit the shell.

It is interesting that the solutions to the left of the knee possess a pattern in their star shapes and propellent distribution. Fig. 7 shows the star shapes which commonly appear in the left of knee region, meaning that these shapes makes a contribution in making the solutions to have a small thrust-match error.

Fig. 8 shows the propellent distribution of a typical solution to the left of knee. An observation reveals that most of these solutions have an almost fixed pattern in the near-core and near-shell layers. This property also contributes in making the solutions to have small thrust-match error.

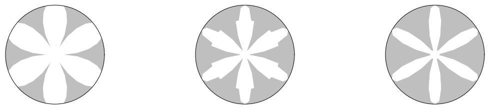


Fig. 7: Star shapes commonly appearing to the left of knee.

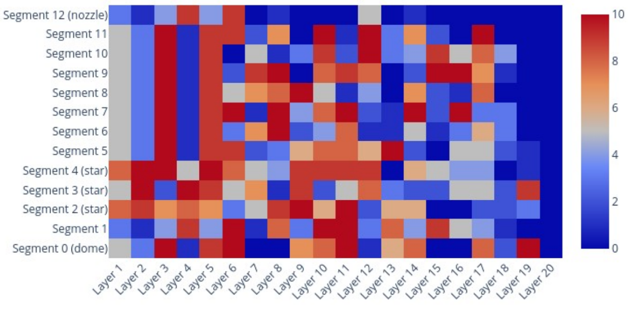
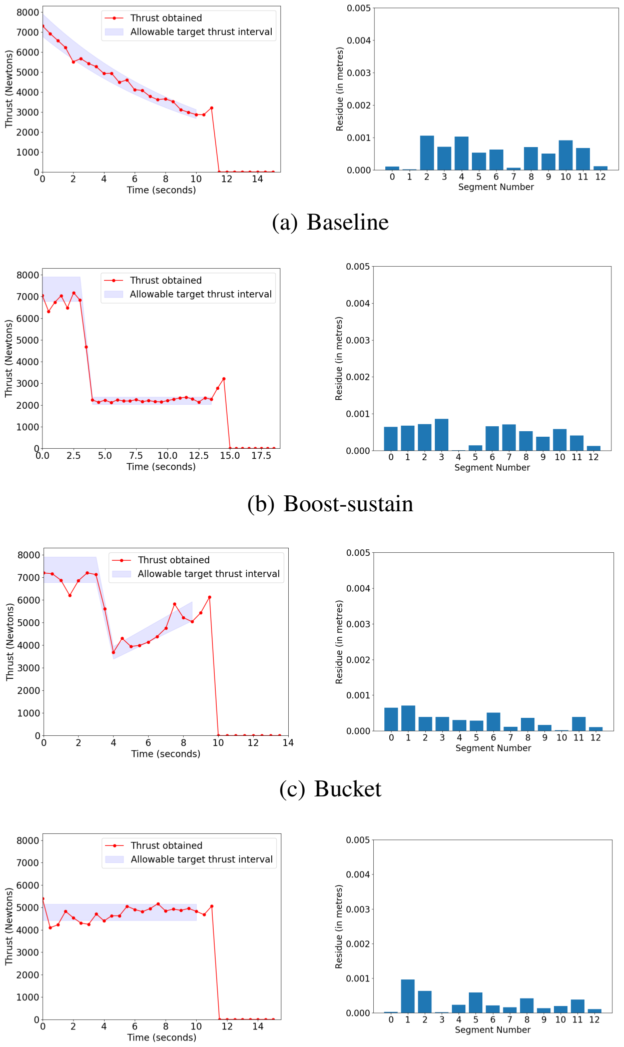


Fig. 8: Propellant distribution for a sample solution to the left of the knee region.

The optimization was run on other thrust profiles as well, and the results for 3 other thrust profiles are shown Fig. 9. It is seen that NSGA-II is able to give residues of the order of 1 mm or less in all the cases along with a good thrust match. All the Pareto Fronts for different thrust profiles show a single knee region with solutions to the left of the knee showing similar properties mentioned earlier. This can be incorporated as a heuristic in the optimization.



(a) Baseline

(b) Boost-sustain

(c) Bucket

(d) Constant thrust

Fig. 9: Thrust profiles and segment-wise residues for different target thrust profiles obtained using base NSGA-II.

V. Innovization-based Optimization Methods

The base NSGA-II optimization is able to give good solutions but at the cost of a large number of function evaluations. The focus of innovization in this problem is to discover rules or principles that reduce the computation cost and achieve faster convergence. The learning process can be based upon human observations, like the approach presented in Section V-A. Conversely, it can also involve machine-discovered rules, like the approach presented in Section V-B. Both the approaches presented in this study focus on reducing the search space by limiting the scope of certain variables.

A. Star Geometry-based Innovization Method

The first innovization approach presented here is based on a manual analysis of the results obtained by the base optimization. Certain shapes like the ones shown in Fig. 7 appear in many solutions on the Pareto-optimal Front for the baseline thrust profile. These shapes can be approximated by smooth curves defined by a smaller number of parameters as compared to the base optimization. The curve used in this study, defined in polar coordinates, is shown in (4). The parameter a defines the star shape. Taking into account the model constraints, $-0.2 \leq a \leq 1.2$ is found to provide a good range of shapes. Fig. 10 shows the star shapes generated by setting a as 0, 0.4 and 0.8.

$$r(a, \theta) = R\,\frac{\sin(\theta + a)}{\sin(\frac{\pi}{6} + a)}, \tag{4}$$

where $0 \leq \theta \leq \frac{\pi}{6}$.

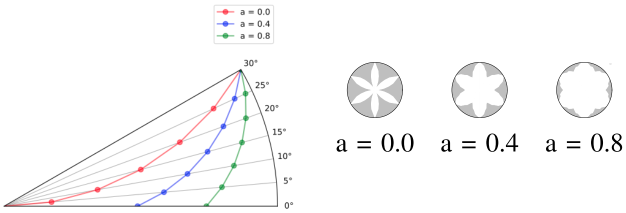


Fig. 10: Sample shapes generated by the simplified star geometry.

Each of the 3 star segments will have a different value of a defining its shape. So instead of 18 variables defining the star, we now have 3 variables. Thus, the number of decision variables now comes out to be 212, down from the 227 variables used in the base optimization. The decision variables for this type of optimization is defined in Table III.

TABLE III: Decision variables for star geometry-based innovization.

Variable	Type	Range	Number
Cylindrical layer propellants	Discrete	[0, 10]	203
Star geometry parameter (a)	Real	$[-0.2, 1.2]$	3
Star section propellants	Discrete	[0, 35]	3
Propellant for circularization	Discrete	[0, 10]	3

B. Adaptive Variable Disabling Innovization Method

In problems involving high numbers of decision variables, such as the rocket design problem, optimization algorithms are faced with a huge search space. For population-based multi-objective methods, after some number of function evaluations, certain variables may converge to fixed values in a majority of the non-dominated solutions. Through the subsequent iterations, the optimization algorithm will waste a lot of computational effort in optimizing over those variables, when it could have focused on optimizing the other variables. This can potentially delay convergence. Thus, some way of telling the optimizer not to consider these variables for subsequent iterations would help in achieving faster convergence. This type of innovization is referred to in the rest of the paper as *Adaptive Disabling Innovization* (ADI). The modified innovization-based NSGA-II optimization algorithm is termed as NSGA-II/ADI.

In this study, the knee-detection procedure presented in [15] was used. The tradeoff metric shown in (5) was used.

$$T(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{m=1}^{M} \max(0, f_m(\mathbf{x}_j) - f_m(\mathbf{x}_i))}{\sum_{m=1}^{M} \max(0, f_m(\mathbf{x}_i) - f_m(\mathbf{x}_j))}, \tag{5}$$

where M is the number of objectives. In (5), $T(\mathbf{x}_i, \mathbf{x}_j)$ represents the total improvement gained per unit deterioration in all the objectives obtained by exchanging solution $\mathbf{x}_j$ with $\mathbf{x}_i$. The quality of a solution $\mathbf{x}_i$ in terms of performance trade-off is evaluated using (6).

$$\mu(\mathbf{x}_i, S) = \min_{j, \mathbf{x}_j \in S, \mathbf{x}_i \not\prec \mathbf{x}_j, \mathbf{x}_j \not\prec \mathbf{x}_i} T(\mathbf{x}_i, \mathbf{x}_j). \tag{6}$$

A larger value of $\mu(\mathbf{x}_i, S)$ within the neighborhood of a solution indicates that it is in a knee region. To identify a knee solution, the following procedure was used.

- Calculate $\mu(\mathbf{x}_i, S)$ for all $\mathbf{x}_i$ in the non-dominated set S according to (6).
- Identify k nearest neighbors for each $\mathbf{x}_i$, where k is set to be 20% of the number of solutions in S.
- Calculate mean and standard deviation for all $\mu(\mathbf{x}_i, S)$.
- If $\mu(\mathbf{x}_i, S)$ is greater than the sum of the mean and half of the standard deviation, the corresponding solution is considered to be a knee solution.

Algorithm 1 NSGA-II/ADI Procedure

```
Input: Population size (N)
Output: Final Pareto-optimal Front ($\mathcal{F}$)
  $gen \leftarrow 1$
  Randomly generate initial population $\mathcal{P}_{gen}$
  while termination condition not satisfied do
      if $(gen \mod 200) = 0$ then
          Detect knee point
          Identify all variables with same value for solutions to the left of knee and make them constants for every solution in $\mathcal{P}_{gen}$
      Create offspring population $\mathcal{Q}_{gen}$ from $\mathcal{P}_{gen}$ using crossover and mutation
      Combine parent and offspring population to create $\mathcal{R}_{gen} = \mathcal{P}_{gen} \cup \mathcal{Q}_{gen}$
      Perform non-dominated sorting on $\mathcal{R}_t$ to identify different fronts $\mathcal{F}_i$ for $i = 1, 2, \ldots,$ and so on
      $\mathcal{P}_{gen+1} \leftarrow \phi$
      $i \leftarrow 1$
      while $|\mathcal{P}_{gen+1}| + |\mathcal{F}_i| < N$ do
          $\mathcal{P}_{gen+1} \leftarrow \mathcal{P}_{gen+1} \cup \mathcal{F}_i$
          $i \leftarrow i + 1$
      Use crowding-sort procedure to select $(N - |\mathcal{P}_{gen+1}|)$ solutions from, $\mathcal{F}_i$, and add them to $\mathcal{P}_{gen+1}$
```

The entire innovization process is presented as Algorithm 1. In this case, the full set of decision variables presented in Table II are used. This means, the star can take up the full range of possible shapes unlike the simplified shapes used in star geometry-based innovization. Once the knee region is detected, the solutions lying on the left of the knee region are analyzed. If all the solutions considered on the left side of the knee region show the same value of certain variables then those variables are treated as constants and do not participate in the optimization during the subsequent generations.

The solutions lying to the left of the knee have many variables that share the same values among solutions. This phenomenon is most notable in the variables defining the star and the propellant material for the first few layers for all segments. However, the values assumed by these variables are not the same for every thrust profile. So even though knowledge about the knee characteristic comes from human observation, an automated method is necessary to decide which variables need to be disabled during the optimization.

C. Results and Discussions for Innovization-based NSGA-II

For comparing the performances of the proposed methods, 20 runs were performed with different random seeds, while ensuring that for each run, the initial population remained the same for all three methods. The optimization was terminated once 40 million function evaluations were completed.

The common star shapes to the left of the knee are shown in Fig. 11. Note that specific star shapes are consistently assigned to the same segments by the optimizer. This is because the three segments have different lengths, so the contribution of a particular star shape depends on the segment in which it is located. It can also be seen that the first few layers of each segment show the same propellant type. This property shows up very early in the optimization and is used by NSGA-II/ADI.

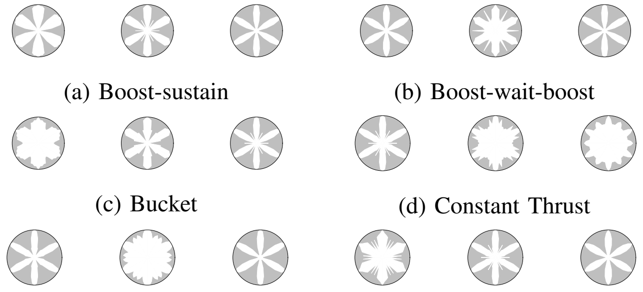
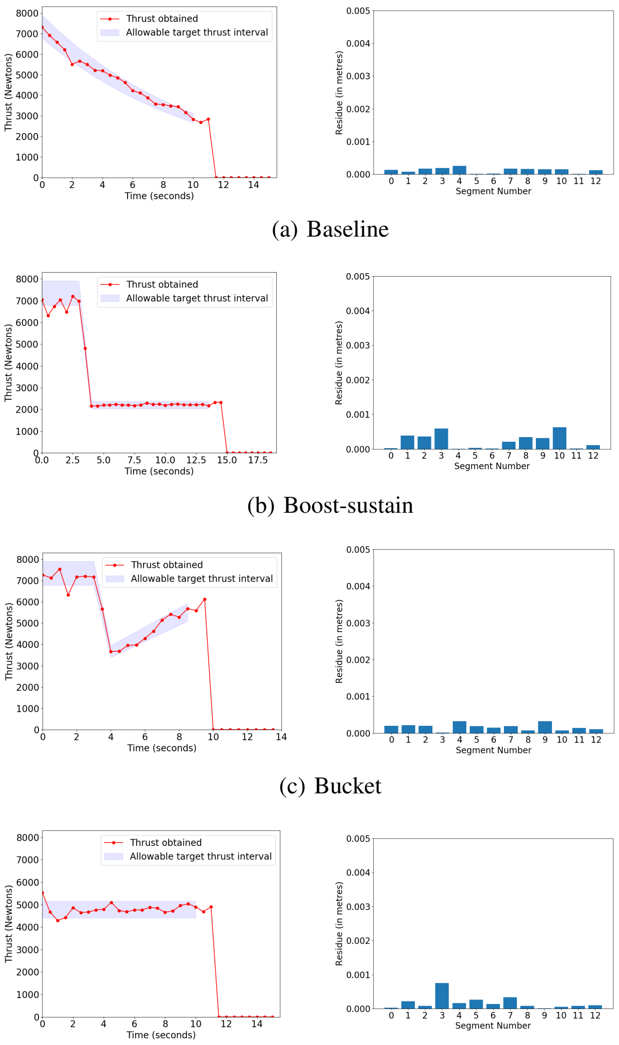

(a) Boost-sustain
(b) Boost-wait-boost
(c) Bucket
(d) Constant Thrust
(e) Hold-regress
(f) Two-step

Fig. 11: Common star geometries in segments 2-4 (left to right).

Thrust and residue plots for the baseline and a few other thrust profiles are shown in Fig. 12. A knee solution was selected as a representative solution in each case.

The Hypervolume (HV) metric [16], [17] has been used to measure the performance of base NSGA-II method and NSGA-II/ADI. HV can measure both convergence and diversity and does not require knowledge of the true Pareto-optimal Front. A higher value of HV indicates better performance. The HV comparison between the base optimization and the two innovization methods is presented for the baseline thrust profile in Fig. 13. For HV calculation, both objectives for Pareto optimal solutions have been normalized between $[0, 1]$. The reference point is then set to be $[1.1, 1.1]$. The average number of variables disabled over the course of the optimization is shown in Table IV.



(a) Baseline
(b) Boost-sustain
(c) Bucket
(d) Constant thrust

Fig. 12: Obtained thrusts and residues for NSGA-II/ADI.

The solid line represents the median HV curve and the shaded region represents the range of HV values in the 20 runs after a particular number of function evaluations. From Fig. 13, it is evident that NSGA-II/ADI is the best performer for the baseline thrust profile. The star geometry-based innovization initially has better HV than the base NSGA-II, but after approximately 10 million function evaluations the rate of improvement slows dramatically, whereas the base NSGA-II keeps on improving. For all thrust profiles, adaptive disabling innovization performs better than the star geometry-based

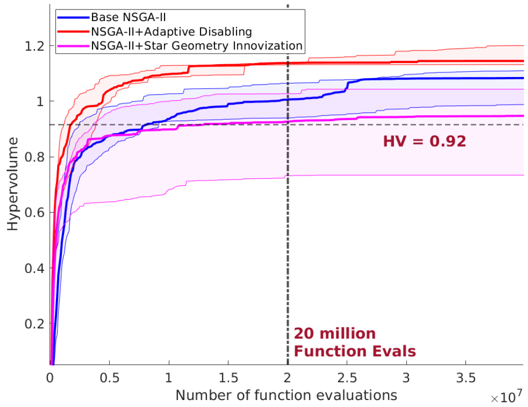


Fig. 13: HV comparison for base NSGA-II and the two innovization methods on the baseline thrust profile.

TABLE IV: Average number of variables disabled over the course of the optimization for each thrust profile.

Thrust Profile	Number of variables disabled	Percentage of variables disabled
Baseline	88	39%
Boost-sustain	83	37%
Boost-wait-boost	79	35%
Bucket	78	34%
Constant Thrust	87	38%
Hold-regress	87	38%
Two-step	76	33%

innovization. This indicates that obtaining good rocket designs through optimization require a search space spanning the full range of star shapes offered by the rocket burn simulator. Hence, for simplicity, only base NSGA-II and NSGA-II/ADI performances have been compared in Fig. 14 for the other thrust profiles. The vertical dashed black line shows the median HV value for both methods after 20 million function evaluations. In all cases, the median HV is better for NSGA-II/ADI compared to the base NSGA-II. Table V shows the median HV values after 20 million function evaluations.

TABLE V: Comparison of median Hypervolume after 20 million function evaluations for all thrust profiles.

Thrust Profile	Median HV after 20 million function evaluations		
	Base NSGA-II	NSGA-II/ADI	Percentage HV improvement
Baseline	1.0070	**1.1380**	▲13.0%
Boost-sustain	0.5466	**0.8038**	▲47.1%
Boost-wait-boost	0.5499	**0.7509**	▲36.6%
Bucket	0.7307	**0.9241**	▲26.5%
Constant Thrust	0.7017	**0.7616**	▲ 8.5%
Hold-regress	0.4417	**0.7793**	▲76.4%
Two-step	0.5957	**0.7349**	▲23.4%

The horizontal dashed line in Fig. 13 and Fig. 14 compares the number of function evaluations taken by each method to achieve a certain HV threshold, which is set as 80% of the best median HV achieved between base optimization and NSGA-II/ADI. The comparison between the two methods is shown in Table VI. It is seen that NSGA-II/ADI achieves a better HV faster than the base NSGA-II, ranging from 43 to 76% savings in function evaluations.

TABLE VI: Approximate number of function evaluations to achieve a median HV Threshold. Percentage savings in overall function evaluations are marked in parenthesis.

Thrust Profile	HV Threshold	Approximate function evaluations (in millions) to reach an HV threshold	
		Base NSGA-II	NSGA-II/ADI
Baseline	0.92	1.3	**0.4** (▼69.2%)
Boost-sustain	0.68	24.0	**6.0** (▼75.0%)
Boost-wait-boost	0.64	30.0	**7.2** (▼76.0%)
Bucket	0.81	9.5	**5.4** (▼43.2%)
Constant Thrust	0.69	17.4	**8.5** (▼51.1%)
Hold-regress	0.75	30.0	**12.0** (▼60.0%)
Two-step	0.70	21.4	**12.0** (▼43.9%)

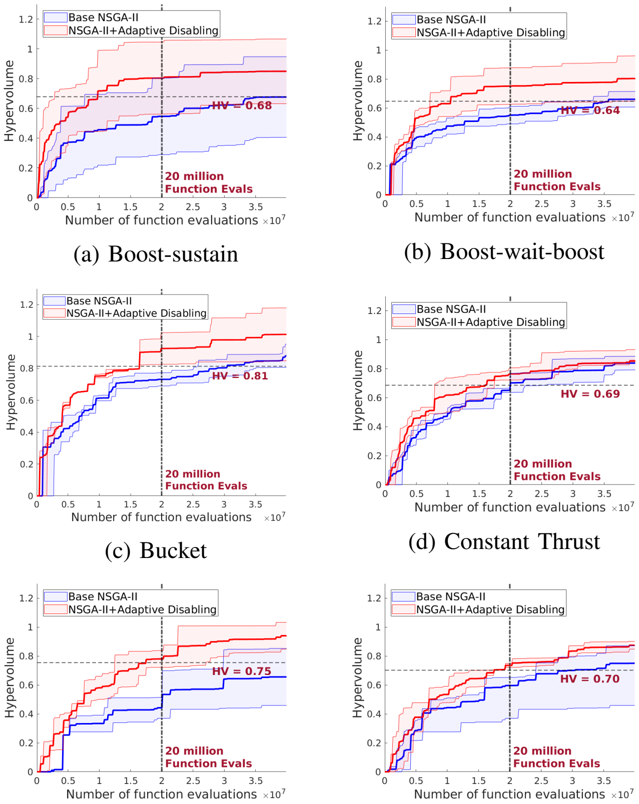
In Fig. 14, the range of HV values represented by the shaded region is very large and often overlaps significantly for base NSGA-II and NSGA-II/ADI. In order to compare the performance, the right-tailed Wilcoxon Signed Rank Test [18], [19] was used. Let x and y represent the final HV values for NSGA-II/ADI and for the base NSGA-II, respectively, over all 20 runs. x and y can be considered as paired observations for each run. The alternate hypothesis tested here states that the difference $x - y$ comes from a distribution with median greater than 0. If true, this means the probability of x having a higher HV than y is high. The hypothesis was tested with 5% significance level and the p-values are shown in Table VII. For all the thrust profiles, the Wilcoxon Right-tailed Signed Rank Test shows a statistically significant improvement in the performance of NSGA-II/ADI over that of base NSGA-II.

TABLE VII: The Wilcoxon Right-tailed Signed Rank Test.

Thrust Profile	p-value	Reject null hypothesis ($p < 0.05$)?
Baseline	0.0125	✓
Boost-sustain	0.0234	✓
Boost-wait-boost	0.0134	✓
Bucket	0.0203	✓
Constant Thrust	0.0065	✓
Hold-regress	0.0083	✓
Two-step	0.0038	✓

VI. Conclusions and Future Work

This study aims to solve a solid fuel rocket motor design problem by representing it as a multi-objective optimization problem with the thrust profile match and simultaneous burnout of fuel throughout the rocket as objectives. NSGA-II was used to solve the problem. In order to reduce the convergence time, two innovization methods were developed. The star geometry-based innovization simplified the optimization problem by reducing the number of decision variables. The adaptive disabling innovization (NSGA-II/ADI) progressively reduces the search space by disabling certain decision variables which converge to fixed values over the course of the optimization. Comparison between the three methods show NSGA-II/ADI to have the best performance, both in terms of solution quality as well as function evaluations for all the thrust profiles. However, it should be noted that the presence of



(a) Boost-sustain

(b) Boost-wait-boost

(c) Bucket

(d) Constant Thrust

(e) Hold-regress

(f) Two-step

Fig. 14: HV comparison between the base NSGA-II and NSGA-II/ADI for different thrust profiles.

heuristics in NSGA-II/ADI runs the risk of leading the search in a sub-optimal direction. A way to improve this is to only modify a percentage of the population selected stochastically, thus maintaining population diversity.

Future work will focus on investigating innovization approaches to discover relationships between individual decision variables. For example, finding if there is any correlation between thrust value at a particular point in time with the propellant material distribution at any location. Such relations can be used back into the optimization for faster convergence. The results obtained in this study provides a good case for performing innovization studies for other real-word large scale problems like truss design optimization.

Acknowledgment

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. We thank Ming Liu and Qiren Gao for some discussions during the development of the optimization algorithm.

References

[1] K. Deb and A. Srinivasan, “Innovization: Innovating design principles through optimization,” in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO ’06. New York, NY, USA: ACM, 2006, pp. 1629–1636. [Online]. Available: http://doi.acm.org/10.1145/1143997.1144266

[2] S. Bandaru and K. Deb, “Automated innovization for simultaneous discovery of multiple rules in bi-objective problems,” in *Evolutionary Multi-Criterion Optimization*, R. H. C. Takahashi, K. Deb, E. F. Wanner, and S. Greco, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1–15.

[3] ——, “Higher and lower-level knowledge discovery from pareto-optimal sets,” *J. of Global Optimization*, vol. 57, no. 2, pp. 281–298, Oct. 2013. [Online]. Available: http://dx.doi.org/10.1007/s10898-012-0026-x

[4] A. Gaur and K. Deb, “Towards an automated innovization method for handling discrete search spaces,” in *2015 IEEE Congress on Evolutionary Computation (CEC)*, May 2015, pp. 2899–2906.

[5] K. Deb and R. Datta, “Hybrid evolutionary multi-objective optimization and analysis of machining operations,” *Engineering Optimization*, vol. 44, no. 6, pp. 685–706, 2012. [Online]. Available: https://doi.org/10.1080/0305215X.2011.604316

[6] A. H. C. Ng, C. Dudas, H. Boström, and K. Deb, “Interleaving innovization with evolutionary multi-objective optimization in production system simulation for faster convergence,” in *Learning and Intelligent Optimization*, G. Nicosia and P. Pardalos, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 1–18.

[7] A. Gaur and K. Deb, “Adaptive use of innovization principles for a faster convergence of evolutionary multi-objective optimization algorithms,” in *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, ser. GECCO ’16 Companion. New York, NY, USA: ACM, 2016, pp. 75–76. [Online]. Available: http://doi.acm.org/10.1145/2908961.2909019

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, April 2002.

[9] J. Blank and K. Deb, “pymoo - Multi-objective Optimization in Python,” https://pymoo.org.

[10] I. Das, “On characterizing the “knee” of the pareto curve based on normal-boundary intersection,” *Structural optimization*, vol. 18, no. 2, pp. 107–115, Oct 1999. [Online]. Available: https://doi.org/10.1007/BF01195985

[11] K. Deb, *Multi-objective Evolutionary Algorithms: Introducing Bias Among Pareto-optimal Solutions*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 263–292. [Online]. Available: https://doi.org/10.1007/978-3-642-18965-4_10

[12] C. A. Mattson, A. A. Mullur, and A. Messac, “Smart pareto filter: obtaining a minimal representation of multiobjective design space,” *Engineering Optimization*, vol. 36, no. 6, pp. 721–740, 2004. [Online]. Available: https://doi.org/10.1080/0305215042000274942

[13] J. Branke, K. Deb, H. Dierolf, and M. Osswald, “Finding knees in multi-objective optimization,” in *Parallel Problem Solving from Nature - PPSN VIII*, X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. E. Rowe, P. Tiňo, A. Kabán, and H.-P. Schwefel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 722–731.

[14] O. de Weck, “Multiobjective optimization: History and promise,” *Invited Keynote Paper, GL2-2, the Third China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems*, vol. 2, 01 2004.

[15] L. Rachmawati and D. Srinivasan, “Multiobjective evolutionary algorithm with controllable focus on the knees of the pareto front,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 810–824, Aug 2009.

[16] D. A. V. Veldhuizen, “Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations,” Evolutionary Computation, Tech. Rep., 1999.

[17] E. Zitzler and L. Thiele, “Multiobjective optimization using evolutionary algorithms — a comparative case study,” in *Parallel Problem Solving from Nature — PPSN V*, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 292–301.

[18] J. D. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 977–979. [Online]. Available: https://doi.org/10.1007/978-3-642-04898-2_420

[19] M. Hollander, D. A. Wolfe, and E. Chicken, *Nonparametric Statistical Methods*. Wiley, Jul. 2015. [Online]. Available: https://doi.org/10.1002/9781119196037