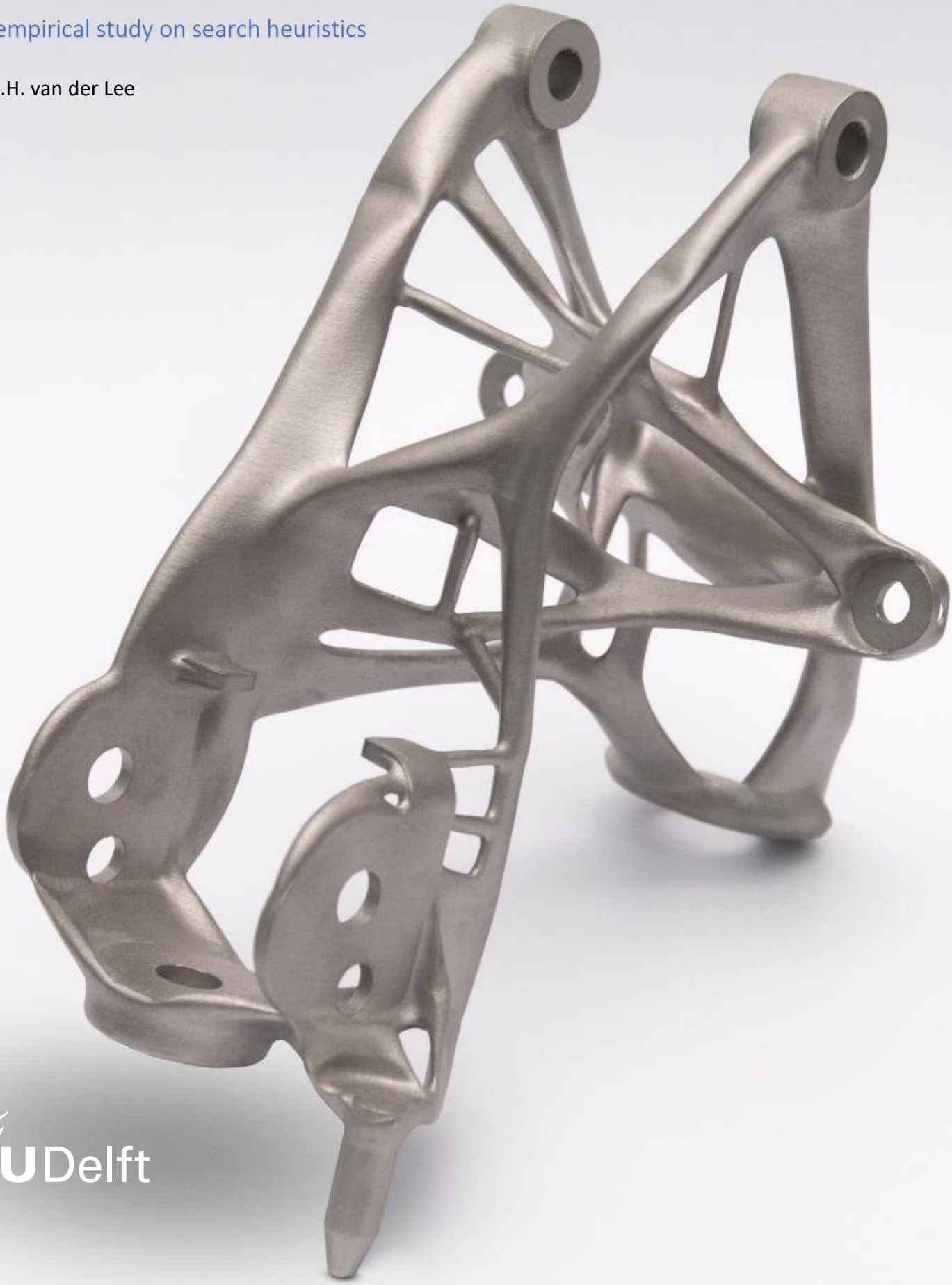
Thickness distribution optimization in flat panels for damage tolerance using genetic algorithms

An empirical study on search heuristics

B.W.H. van der Lee



TU Delft

Keywords; computational design, fatigue damage tolerance, crenellation, genetic algorithms, evolutionary computing, crossover, mutation, search heuristics, engineering design, fuselage

THICKNESS DISTRIBUTION OPTIMISATION IN FLAT PANELS FOR DAMAGE TOLERANCE USING GENETIC ALGORITHMS

By

B. W. H. van der Lee

in partial fulfilment of the requirements for the degree of

Master of Science

in Aerospace Engineering

at the Delft University of Technology,

to be defended publicly on Tuesday June 30th, 2020 at 10:00 AM.

Supervisor:	Dr. C.D. (Calvin) Rans,	TU Delft
Thesis committee:	Dr.-Ing. S.G.P. (Saullo) Castro,	TU Delft
	Dr. S.R. (Sergio) Turteltaub,	TU Delft

Preface

In front of you lies a thesis which aims to spark a thorough discussion on the application of genetic algorithms in engineering design. This thesis was foremostly written for readers wishing to understand the considerations in applying genetic algorithms to (engineering) design problems. From my personal experience it is not easy to find critical work on this topic in literature and I hope it helps you in evaluating your methodology.

Having studied for an engineering degree for several years, the field of machine learning and evolutionary algorithms was nothing like the calculus-based mathematics from all my courses. Furthermore, the field of search heuristics turned out to be overwhelmingly vast while comparing methodology and theories from previous research was often made very complex. It was not uncommon to miss important details, substantiation of methodology or naming conventions - something which really shocked me as a student whose course books always provided some basis of scientific understanding.

Nonetheless, in the past years, I have often found myself searching for simple ways to investigate the behaviour of genetic algorithms on my design problem, as well as how to effectively communicate to readers what I uncovered about their mechanisms. While this field of research has definitely not yet matured and many questions remain to be asked, I do hope that these few insights can at least convince researchers and practitioners to not accept the standard genetic algorithm methodology at face value.

Foremost, I would like to thank Calvin for this endless patience with my research. As Calvin would sometimes say, *if you would have loved to have read your own thesis before you started, then you have succeeded*. From that perspective I feel proud to say that I wish I had. I admire your teaching philosophy and hope you will continue to inspire many others for a long time.

Lastly, if it weren't for my loving friends and family, I might not have been able to conclude my MSc thesis. I would like to thank my dear friend Jobbe, who took his free time to hear my story in all detail and with whom I could refine the many ideas I had. Nina, who welcomed me to her home closer to Delft for over a month in order finish "the last mile" of my thesis. Rutger and Alet, who have been unbelievably generous with their attention while helping me remain focussed on the end goal. My father, who has unconditionally supported me during my entire education, as well as my mother who always knows what I need to hear. Giguru, Ivar and Bibianne, who were there for providing a listening ear and asking critical questions.

Last but not least, I most grateful for my partner Goda who has been both very understanding as well as consistently pushing me to finish the work in this very long journey. Thank you so much for your dedication.

Finally, I am grateful to myself. I have never felt so challenged to persist in the face of many personal doubts and insecurities. The life lessons I have taken from this journey are to stay true to your heart and dare to ask for help in confronting whatever is blocking you. If not you, then no one will do it for you.

Bart van der Lee Amsterdam, June 22nd , 2020

ABSTRACT

Additive manufacturing is a collection of concepts using a bottom-up, layer-by-layer approach and may have many advantages in fabricating more lightweight and durable component design (Roy, et al., 2008). Lighter and more durable components could enable more energy efficient travel and contribute greatly to environmental protection of the earth or deep space travel.

However, present day design methods are unsuitable for the complex nature of optimally designing such detailed and organic shapes (Marler & Arora, 2004). As such, stochastic optimisation techniques such as the standard genetic algorithm (SGA) have been applied in numerous engineering design cases, initially yielding promising yet strongly varying results. These varying, unreliable results are limiting the broader application of such stochastic techniques in engineering design, which requires clear explanation of a design method.

The inexplicability of experimental results is caused by a very poor understanding of how such stochastic optimisation process work (Sorensen, et al., 2017). This research used a simple design case study of thickness distribution on a flat fuselage plate to investigate the mechanisms through which a GA interacts on such problem. The fuselage design was optimised for fatigue damage tolerance per unit weight on various levels of granularity using an adapted analytical model.

To analyse the optimisation process, a functional definition of what GAs “should be doing” was determined. These GA functions were used to evaluate and compare different heuristic designs. Furthermore, a sensitivity study was performed on the mutation rate and initial population of the GA.

The results of this study provided evidence that the standard genetic algorithm (SGA) design of crossover and random mutation is unreliable, inefficient and deceptive, while an improved GA (IGA) design using concepts such as signalling of relative allele strength, mutation filtering and selection bias can create a reliable, better performing and understandable optimisation process in the context of the design case.

Furthermore, a broader discussion on the results demonstrated that standard crossover and the improved mutation are inherently more similar than we expected, thus questioning whether it is not more important to design a set of search heuristics through better understanding of the fitness space, rather than the application of a flawed, nature-inspired standard crossover and random mutation. This research questions whether we should name these heuristics crossover and mutation in the first place.

Through these insights, this research contributed to ongoing research in understanding GAs, which, if better understood, could assist engineers in finding improved designs of additively manufactured components.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY OF ABBREVIATIONS

Table 1 Overview of abbreviations and their description grouped by whether they are found in literature

	Abbreviation	Meaning
Found in literature	GA	Genetic algorithm
	SGA	Standard Genetic Algorithm
	IGA	Improved Genetic Algorithm
Not found in literature	PA	Positive allele
	NA	Negative allele
	PRAS	Population Relative Allele Strength
	PRASF	Population Relative Allele Strength over Frequency

LIST OF EXPERIMENTS

Table 2 Experiment matrix with all the experiments documented in the thesis report

EXPID	Changing Variable*	Selection Operator	Initialization Operator	Crossover Operator	Mutation Operator	PRAS signalling	Mutation Filtering	Selection Bias	Allele Enumeration	Mutation Rate	Objective Function	Granularity	Initial PRAS	Npop	Generations	DEXPID**
EX-1	Crossover	T-S	RD-I	SP-C						0.1	N/A^m	30	mid	10	20	238
EX-2	Crossover	T-S	RD-I	TP-C						0.1	N/A^m	30	mid	10	20	239
EX-3	Crossover	T-S	RD-I	U-C						0.1	N/A^m	30	mid	10	20	240
EX-4	Initial PRAS	T-S	DD-I	U-C						0.1	N/A^m	30	low	10	20	313
EX-5	Initial PRAS	T-S	DD-I	TP-C						0.1	N/A^m	30	low	10	20	314
EX-6	Initial PRAS	T-S	DD-I	U-C						0.1	N/A^m	30	high	10	20	315
EX-7	Initial PRAS	T-S	DD-I	TP-C						0.1	N/A^m	30	high	10	20	316
EX-8	Initial PRAS	T-S	DD-I	SP-C						0.1	N/A^m	30	high	10	20	317
EX-9	Initial PRAS	T-S	DD-I	SP-C						0.1	N/A^m	30	low	10	20	319
EX-10	Granularity	T-S	RD-I	U-C						0.1	N/A^m	150	mid	10	20	284
EX-11	Mutation	T-S	RD-I		U-M					0.1	N/A^m	30	mid	10	20	176
EX-12	Mutation	T-S	RD-I		RS-M	X				0.1	N/A^m	30	mid	10	20	178
EX-13	Mutation	T-S	RD-I		RSM-M	X				0.1	N/A^m	30	mid	10	20	179
EX-14	Mutation	T-S	RD-I		RSAA-M	X	X			0.1	N/A^m	30	mid	10	20	182
EX-15	Mutation	T-S	RD-I		RSAASS-M	X	X	X		0.1	N/A^m	30	mid	10	20	258
EX-16	Mutation	T-S	RD-I		RSAASSEUE-M	X	X	X	X	0.1	N/A^m	30	mid	10	20	289
EX-17	Mutation	T-S	RD-I	U-C	U-M					0.1	N/A^m	30	mid	10	20	147
EX-18	Mutation	T-S	RD-I	U-C	RSM-M	X				0.1	N/A^m	30	mid	10	20	189
EX-19	Mutation	T-S	RD-I	U-C	RS-M	X				0.1	N/A^m	30	mid	10	20	135
EX-20	Mutation	T-S	RD-I	U-C	RSAA-M	X	X			0.1	N/A^m	30	mid	10	20	183
EX-21	Mutation	T-S	RD-I	U-C	RSAASS-M	X	X	X		0.1	N/A^m	30	mid	10	20	270
EX-22	Mutation	T-S	RD-I	U-C	RSAASSEUE-M	X	X	X	X	0.1	N/A^m	30	mid	10	20	287
EX-23	Initial PRAS	T-S	DU-I	U-C	RSAASSEUE-M	X	X	X	X	0.1	N/A^m	30	low	10	20	290
EX-24	Initial PRAS	T-S	DU-I	U-C	U-M					0.1	N/A^m	30	low	10	20	291
EX-25	Initial PRAS	T-S	DU-I	U-C	RS-M	X				0.1	N/A^m	30	low	10	20	292
EX-26	Initial PRAS	T-S	DU-I	U-C	RSAA-M	X	X			0.1	N/A^m	30	low	10	20	298
EX-27	Initial PRAS	T-S	DU-I	U-C	RSAASS-M	X	X	X		0.1	N/A^m	30	low	10	20	300
EX-28	Initial PRAS	T-S	DD-I	U-C	RS-M	X				0.1	N/A^m	30	high	10	20	302
EX-29	Initial PRAS	T-S	DD-I	U-C	RS-M	X				0.1	N/A^m	30	low	10	20	303
EX-30	Initial PRAS	T-S	DD-I	U-C	RSAA-M	X	X			0.1	N/A^m	30	low	10	20	297
EX-31	Initial PRAS	T-S	DD-I	U-C	RSAA-M	X	X			0.1	N/A^m	30	high	10	20	304
EX-32	Initial PRAS	T-S	DD-I	U-C	RSAASS-M	X	X	X		0.1	N/A^m	30	low	10	20	305
EX-33	Initial PRAS	T-S	DD-I	U-C	RSAASS-M	X	X	X		0.1	N/A^m	30	high	10	20	306
EX-34	Initial PRAS	T-S	DD-I	U-C	RSAASSEUE-M	X	X	X	X	0.1	N/A^m	30	high	10	20	306

EX-35	Initial PRAS	T-S	DD-I	U-C	RSAASSEUE-M	X	X	X	X	0.1	N/A^m	30	low	10	20	307
EX-36	Initial PRAS	T-S	DD-I	U-C	U-M					0.1	N/A^m	30	low	10	20	308
EX-37	Initial PRAS	T-S	DD-I	U-C	U-M					0.1	N/A^m	30	high	10	20	309
EX-38	Initial PRAS	T-S	DU-I	U-C	RSAASSEUE-M	X	X	X	X	0.2	N/A^m	30	low	10	20	310
EX-39	Initial PRAS	T-S	DU-I	U-C	RS-M	X				0.2	N/A^m	30	low	10	20	311
EX-40	Initial PRAS	T-S	DU-I	U-C	U-M					0.2	N/A^m	30	low	10	20	312
EX-41	Mutation Rate	T-S	RD-I	U-C	RS-M	X				0.05	N/A^m	30	mid	10	20	227
EX-42	Mutation Rate	T-S	RD-I	U-C	RSAA-M	X	X	X		0.05	N/A^m	30	mid	10	20	231
EX-43	Mutation Rate	T-S	RD-I	U-C	RSAASS-M	X	X	X		0.05	N/A^m	30	mid	10	20	259
EX-44	Mutation Rate	T-S	RD-I	U-C	U-M					0.05	N/A^m	30	mid	10	20	276
EX-45	Mutation Rate	T-S	RD-I	U-C	RSAASSEUE-M	X	X	X	X	0.05	N/A^m	30	mid	10	20	321
EX-46	Mutation Rate	T-S	RD-I	U-C	RS-M	X				0.1	N/A^m	30	mid	10	20	228
EX-47	Mutation Rate	T-S	RD-I	U-C	RSAA-M	X	X			0.1	N/A^m	30	mid	10	20	232
EX-48	Mutation Rate	T-S	RD-I	U-C	RSAASS-M	X	X	X		0.1	N/A^m	30	mid	10	20	266
EX-49	Mutation Rate	T-S	RD-I	U-C	U-M					0.1	N/A^m	30	mid	10	20	277
EX-50	Mutation Rate	T-S	RD-I	U-C	RSAASSEUE-M	X	X	X	X	0.1	N/A^m	30	mid	10	20	322
EX-51	Mutation Rate	T-S	RD-I	U-C	RS-M	X				0.15	N/A^m	30	mid	10	20	229
EX-52	Mutation Rate	T-S	RD-I	U-C	RSAA-M	X	X			0.15	N/A^m	30	mid	10	20	233
EX-53	Mutation Rate	T-S	RD-I	U-C	RSAASS-M	X	X	X		0.15	N/A^m	30	mid	10	20	267
EX-54	Mutation Rate	T-S	RD-I	U-C	U-M					0.15	N/A^m	30	mid	10	20	278
EX-55	Mutation Rate	T-S	RD-I	U-C	RSAASSEUE-M	X	X	X	X	0.15	N/A^m	30	mid	10	20	323
EX-56	Mutation Rate	T-S	RD-I	U-C	RS-M	X				0.2	N/A^m	30	mid	10	20	230
EX-57	Mutation Rate	T-S	RD-I	U-C	RSAA-M	X	X			0.2	N/A^m	30	mid	10	20	234
EX-58	Mutation Rate	T-S	RD-I	U-C	RSAASS-M	X	X	X		0.2	N/A^m	30	mid	10	20	268
EX-59	Mutation Rate	T-S	RD-I	U-C	U-M					0.2	N/A^m	30	mid	10	20	279
EX-60	Mutation Rate	T-S	RD-I	U-C	RSAASSEUE-M	X	X	X	X	0.2	N/A^m	30	mid	10	20	320
EX-61	Granularity	T-S	RD-I	U-C	RSAA-M	X	X			0.1	N/A^m	150	mid	10	20	186
EX-62	Granularity	T-S	RD-I	U-C	RSAASS-M	X	X	X		0.1	N/A^m	150	mid	10	20	187
EX-63	Granularity	T-S	RD-I	U-C	RS-M	X				0.1	N/A^m	150	mid	10	20	272
EX-64	Granularity	T-S	RD-I	U-C	U-M					0.1	N/A^m	150	mid	10	20	273
EX-65	Granularity	T-S	RD-I	U-C	RSAASSEUE-M	X	X	X	X	0.1	N/A^m	150	mid	10	20	288
EX-66	Generations	T-S	RD-I	U-C	RSAASS-M	X	X	X		0.1	N/A^m	150	mid	10	40	275
EX-67	Generations	T-S	RD-I	U-C	RSAASSEUE-M	X	X	X	X	0.1	N/A^m	150	mid	10	40	325

*Changing variable with respect to the preceding experiments in this list, improves understandability of the experimentation matrix

** Experiment ID in database with all experiments. Experiment list shown in this report contains a fraction of total number of experiment run

1. INTRODUCTION

The aircraft and space industry continue to seek lighter and more durable designs of structures or components. Traditional subtractive and formative manufacturing techniques have physical limits in achieving more lightweight designs of aircraft components, while larger structures often requiring joining methods such as fastening and bonding.

Additive manufacturing, on the other hand, is a collection of concepts using a layer-by-layer approach and may have many advantages in fabricating lightweight component and structure design. Even lighter components could enable more energy efficient travel which could facilitate further deep space travel and a reduced environmental pressure.

While efforts in overcoming the manufacturing and quality challenges are being done, the close to unlimited design freedom to create more organic shapes has still been limited by traditional design methods (Yang & Zhao, 2015). Therefore, optimisation algorithms are necessary to support humans in designing more complex, natural forms where the design variables n are in the range $n > 100$.

Due to the complex nature of these design problems, gradient-based optimization methods are generally not applicable nor effective due to smooth and continuous gradient information being absent or very difficult to obtain. Most of these problems can be formulated as combinatorial optimisation problems (ElMaraghy, et al., 2012).

Genetic algorithms (GA), a class of evolutionary algorithms, have been found useful heuristics on such problems in engineering design applications, specifically in thickness distribution optimisation. For example, a butterfly valve disc thickness distribution design the maximum stress was decreased with 61% and maximum flow was increased with 17%, while mass was reduced by 6%, as compared to an initial manufacturer design (Caraballo, et al., 2017). Another example is that of crenellation of fuselage plates, where optimized designs showed a 10% fatigue life improved as compared to an initial design that was developed based on experience (Lu, et al., 2016).

However, there is still discrepancy between theory and empirical observations in GA applications in higher ranges of n, particularly because no empirical basis seems to exist for a generally accepted theoretical description (Sorensen, et al., 2017). Such scale is necessary, as small ranges of n would imply a limited design freedom, thus removing the potential benefits of additive manufacturing. This discrepancy makes the optimal design of GAs and the predictability of their functioning difficult to achieve. Improving the understanding of GAs mechanisms could significantly improve the design possibilities and thus allow us to benefit from additive manufacturing and many other bottom-up design approaches.

Empirical evidence supports that variations of GAs can have a large influence on the results, both positively and negatively (Wolpert & Macready, 1997). As such, the arbitrary choice of the SGA with standard heuristics, arbitrary choice of parameters and objective functions would be inappropriate. As far as we know, no empirical information has been published on the effect of GA heuristic variations and parameters on the

thickness distribution optimisation of flat fuselage plates, as well as the influence of objective function variations.

More generally, wide acceptance of GAs in engineering design has not yet been established due to this lack of understanding GAs. To enable broader acceptance in engineering design, it is of great importance that we create a better understanding of the mechanisms of the GA.

The objective of this research is to improve reliability (expressed as the probability that the GA is able to find known, optimal solutions) in thickness distribution optimization models of crenellated aircraft fuselage plates by (1) improving the standard genetic algorithms and (2) finding suitable objective functions. Specifically, these objectives will be tested in the range of $t = 3$, $3 < n < 150$.

We expect the trial-and-error exploration of different GA heuristics to contribute valuable information on how the GA mechanisms work on a simple design case study of crenellation and provide a basis for further understanding the fascinating interaction between evolutionary algorithms and engineering design.

2. THEORETICAL FRAMEWORK

In this chapter the reader is introduced to existing engineering models for the thickness distribution optimisation of flat plates, as well as the standard genetic algorithm.

First, we introduce the *phenomenon of crenellation*, as it was shown to enhance the fatigue tolerance of metallic plates. To the best of our knowledge, no analytical model to describe this phenomenon exists.

Therefore, second, to develop such model, *an analytical bonded-stiffener fatigue crack growth model* (Rans, et al., 2015) developed by Rans, Rodi and Alderliesten is presented which will be adapted in the methods section in an attempt to describe fatigue crack growth for crenellated metal plates. By doing so, it would become possible to define a simplified design case study of crenellation upon which a genetic algorithm could be studied in detail.

Third, as for the optimisation model for this research, the *standard genetic algorithm (SGA*), also referred to as the canonical genetical algorithm (CGA), is introduced in order to build a common, basic understanding of the genetic algorithm.

2.1. CRENELLATION

2.1.1. THEORY

Crenellations are one-sided, systemic variations in plate thickness (Fig 1b) and were first studied as a new design philosophy in skin and stringer geometries on laser beam welded (LBW) stiffened, Al2139-T8, large centre cracked flat panels by Uz et al in 2008 (Uz, et al., 2009). The idea was to modify the stress intensity factor (SIF) distribution along the crack growth period by which the cumulative fatigue life was expected to improve without any additional weight increase.

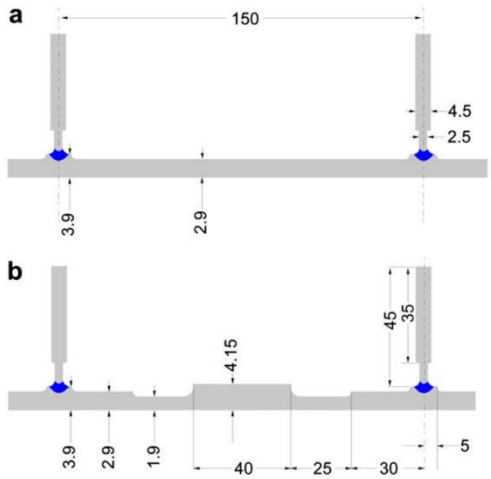


Figure 1 Flat panel and crenellated panel design (Uz, et al., 2009)

The mechanism has been mathematically described by Uz et al (Uz, et al., 2009) as follows.

Consider a cracked, metallic plate for which the fatigue crack propagation (FCP) behaviour can be described by the Paris Law $\left(\frac{da}{dN} = C\Delta K^m ; \Delta K = K_{max} - K_{min}\right)$. Constant amplitude loading with a cyclic far field stress creates a stress intensity factor (SIF) range ΔK_a at the crack tip. If this SIF range is increased for half of the crack length and reduced with the same factor for the other half, it seems to be possible to increase the cumulative fatigue life compared to when the crack is driven by the original SIF range for the entire length.

For example, the lifetime in stress cycles N of a crack driven by SIF range of ΔK_a over a small crack increment of $2\ \Delta a$ can be determined as

$$N_{ref} = \frac{2\Delta a}{C\Delta K_a^m} \quad (1)$$

assuming that the crack increment of $2\Delta a$ is short enough, such that the increase in SIF as a result of the incremental increase crack length is negligible.

Also neglecting any loading history effects, if it is possible to reduce ΔK_a by a factor t for the first half of the crack increment, and increase it with the same factor for the second half, the modified fatigue life is then described as

$$N_{mod} = \frac{\Delta a}{C(\Delta K_a - t\Delta K_a)^m} + \frac{\Delta a}{C(\Delta K_a + t\Delta K_a)^m} \quad (2)$$

As such, the hypothesis was formed that for any value of t in the interval (0,1) the modified fatigue life exceeds the reference value

$$N_{mod} > N_{ref} \quad (3)$$

Details for the examination of this inequality are given in the Appendix. The argumentation above seems to indicate that fatigue life gain in the slow growth region will be higher than the life shortening in the fast growth region as a result of modifications in the SIF range. To understand the expected phenomenon in more detail, let us consider the available empirical data.

2.1.2. EMPIRICAL EVIDENCE

Empirical evidence has shown that the shape of the SIF distribution as a result of crenellation is rather more complex than the described step-function above. Crenellated panels showed significantly improved fatigue lives compared to flat reference panels under constant amplitude loading (Fig 2). Under variable amplitude loading, the improvement increased further due to more efficient crack retardation in thinner regions of the crenellated panel (figure added to appendix) (Uz, et al., 2009).

Simple variations have empirically shown to improve the overall fatigue life and buckling performance per unit weight (Quinn, et al., 2011).

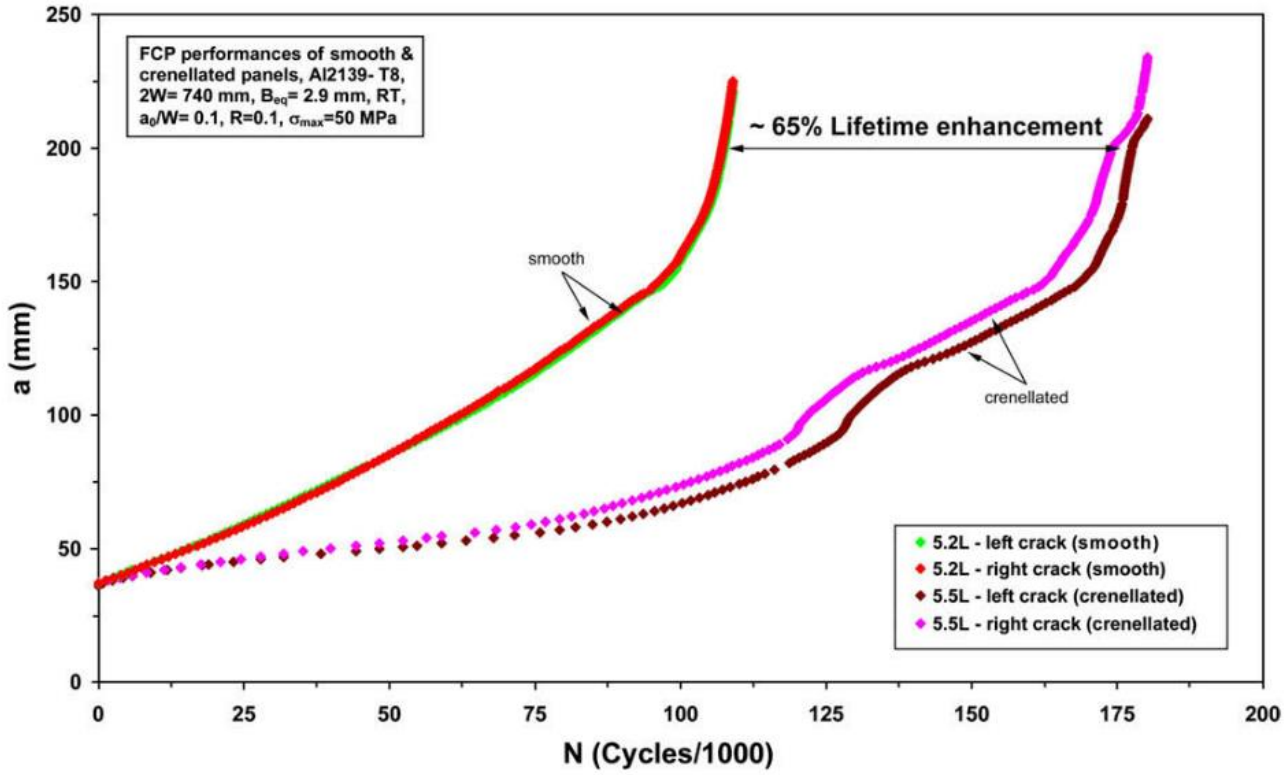


Figure 2 Comparison of a vs N curves of reference and crenellated panels under constant amplitude loading (Uz, et al., 2009)

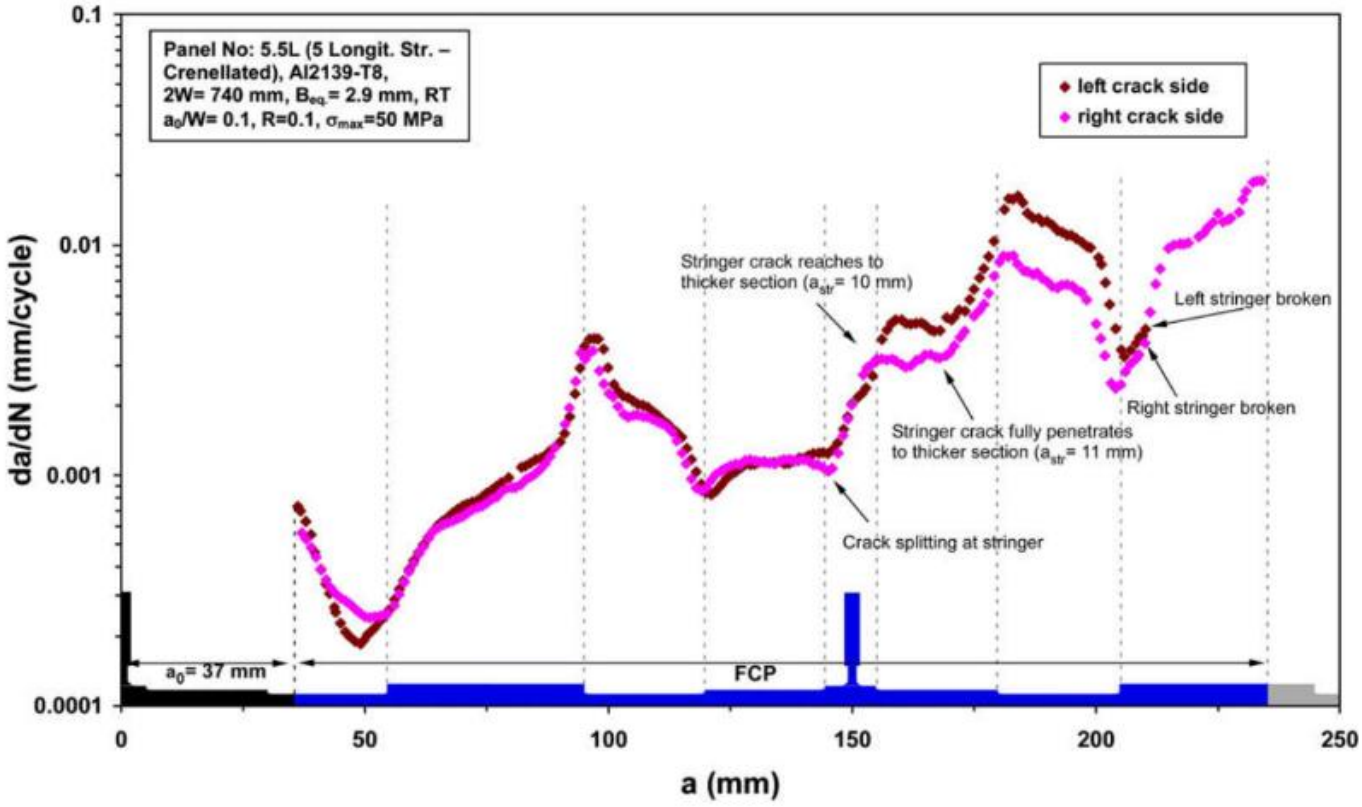


Figure 3 Experimental crack growth rates for varying crack lengths on a crenellated plate under constant amplitude loading (Uz, et al., 2009)

As the design concept of crenellation offers possibilities for tailoring of the damage tolerance attributes of objects, the concept may be used to develop “optimal crenellation patterns” for different parts of the airframe with the accompanying weight savings. The overarching goal is to find a suitable method to optimise the problem of crenellation design.

For any optimisation to take place, the natural next step would be to consider ways in which the phenomenon can be captured in a model. For this, the next section of this chapter introduces an analytical model which will be adapted in this research to model a simplified crack growth response under crenellation patterns.

2.2. FATIGUE CRACK GROWTH MODEL FOR BONDED STIFFENERS

2.2.1. THEORY

Theoretically the number of unique solutions using a bottom-up manufacturing approach as additive manufacturing can become near to indefinite. Manufacturing and testing of every design solution is not feasible. Therefore, the possibility to design very efficient structures is determined by how accurately and computationally inexpensively we can model (thus predict) the failure behaviour of a very large range of design solutions. Analytical models, though much more difficult to find, offer such an inexpensive and simple way to evaluate design solutions. If such an overarching model can accurately predict failure in many different failure modes (fatigue life, but also buckling, torsion, bending, etc), optimisation algorithms would be able to search the entire solution space for the most efficient designs depending on de designer objectives with complex trade-offs and constraints.

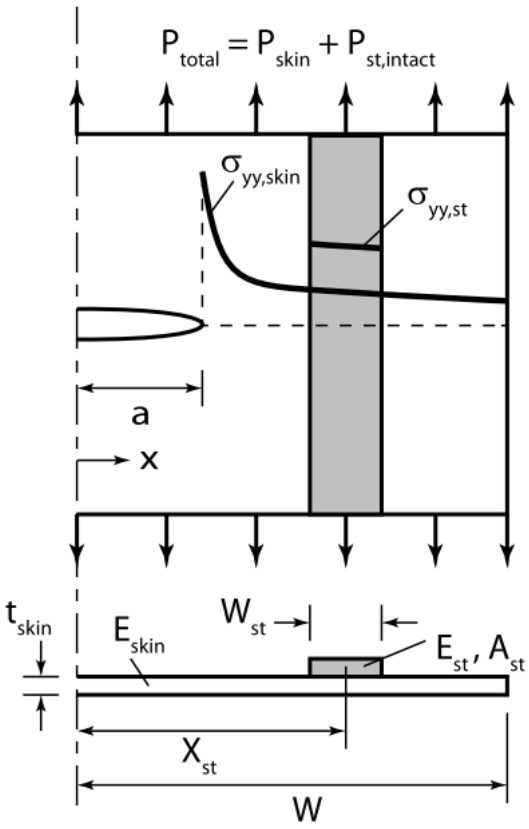


Figure 4 Westegaard stress distribution for an intact stiffener ahead of the crack tip (Rans, et al., 2015)

For the scope of this research, the evaluation of design solutions is limited to fatigue damage tolerance where the thickness can be redistributed along the plate width. To model the phenomenon of crenellation (varying 2D thickness) in metallic plates in terms of fatigue damage, an analytical model for the prediction of crack growth for cracked panels containing bonded stiffeners by Rans et al is adapted (Rans, et al., 2015). The model is compactly presented below, for the interested reader the full paper can be found in the references.

In this case, the general approach in modelling crack growth in a metallic, crenellated plate relies on linear elastic fracture mechanics (LEFM) and Mode I stress intensity factors. The model decomposes the bonded stiffened panel into three stiffener conditions,

- Intact stiffeners ahead of the crack tip. These stiffeners provide additional stiffness ahead of the crack tip, thus reducing crack growth.
- Broken stiffeners behind the crack tip. These stiffeners transmit load into the panel along the crack flanks, thus increasing crack growth.
- Bridging stiffeners over the crack. These stiffeners bridge load over the crack, thus reducing crack growth.

By dividing the complex cracked stiffened panel into these components (see Appendix B. for a visual of these components), the SIF for each of the more simple conditions could be developed. The overall SIF for the stiffened crack panel can then be found through addition of the components.

SIF DUE TO INTACT STIFFENER AHEAD OF THE CRACK

Consider a symmetric cracked panel with a single bonded stiffener ahead of the crack tip, as shown in Figure 4 Westegaard stress distribution for an intact stiffener ahead of the crack tip Figure 4. Furthermore, assume that the stress state in the panel ahead of the crack tip follows the Westergaard stress distribution

$$\sigma_{yy,skin} = \frac{\sigma_{skin}}{\sqrt{1-\left(\frac{a}{x}\right)^2}} \quad for\ \mathrm{a} \leq \mathrm{x} \leq \mathrm{W} \tag{4}$$

Where x is the distance from the crack centre along the crack plane and σ_{skin} the far field stress carried by the panel in the cracked condition, determined as

$$\sigma_{skin} = \frac{P_{skin} - F_{intact}}{W \cdot t_{skin}} \tag{5}$$

where the σ_{skin} is unknown in Eq 2 since the value of F_{intact} is unknown. This model solves this issue through the assumption of an isostrain condition between the stiffener and skin panel. Thereafter, it follows for the stiffener

$$\sigma_{yy,st} = \frac{E_{st}}{E_{skin}} \cdot \frac{\sigma_{skin}}{\sqrt{1-\left(\frac{a}{x}\right)^2}} \quad for\ \left(X_{st} - \frac{W_{st}}{2}\right) \leq \mathrm{x} \leq \left(X_{st} + \frac{W_{st}}{2}\right) \tag{6}$$

The total load carried by the stiffened panel in Figure 4 is equivalent to the sum of the far field loads in the skin and the intact stiffener. As such, the integral of the stress distributions along the skin $\sigma_{yy,skin}$ (Eq 1) and the stiffener $\sigma_{yy,st}$ (Eq 3) must be in equilibrium with the far field applied loads $P_{skin} + P_{st,intact}$.

$$P_{skin} + P_{st,intact} = \int_a^W \sigma_{yy,skin} \cdot t_{skin} \cdot dx + \int_{W_{st}} \sigma_{yy,st} \cdot \left(\frac{A_{st}}{W_{st}}\right) \cdot dx \quad (7)$$

Rearranging for σ_{skin}

$$\sigma_{skin} = \frac{P_{skin} + P_{st,intact}}{\int_a^W \frac{t_{skin}}{\sqrt{1-\left(\frac{a}{x}\right)^2}} \cdot dx + \int_{W_{st}} \left(\frac{E_{st}}{E_{skin}}\right) \cdot \left(\frac{A_{st}}{W_{st}}\right) \frac{1}{\sqrt{1-\left(\frac{a}{x}\right)^2}} \cdot dx} \quad (8)$$

Following the definition of the stress intensity factor, the value of K_{intact} with n number of intact stiffeners ahead of the crack tip in terms of the far field stress from a Westergaard distribution can be written as follows,

$$K_{intact} = \sigma_{skin} \sqrt{\pi a} = \frac{\left(P_{skin} + \sum_{i=1}^{n} \left(P_{st,intact}\right)_i\right) \sqrt{\pi a}}{\int_a^W \frac{t_{skin}}{\sqrt{1-\left(\frac{a}{x}\right)^2}} \cdot dx + \sum_{i=1}^{n} \left[\int_{W_{st}} \left(\frac{E_{st}}{E_{skin}}\right) \cdot \left(\frac{A_{st}}{W_{st}}\right) \frac{1}{\sqrt{1-\left(\frac{a}{x}\right)^2}} \cdot dx\right]_i} \quad (9)$$

The described model provides a way to describe crack growth in terms of fluctuating plate thickness, with the largest difference with crenellation being that the materials are the same. Assuming that this model can be adapted to describe crenellation and thus evaluate any given design solution, a method to heuristically optimize the crenellation pattern would have been found.

2.3. STANDARD GENETIC ALGORITHM

2.3.1. GENETIC ALGORITHM

The genetic algorithm is a population-based, stochastic heuristic optimisation method inspired by natural evolution. The idea is to improve the value of a set of solutions in terms of the objective function by iteratively introducing forms of random variation, random exchange of information and selection pressure.

At every iteration, called a generation, a population of solutions exchange attributes using rules for exchange of information, called crossover, and variation, called mutation. This new set of solutions is then subjected to selection pressure, where weak solutions are removed from the population. While it can be empirically

shown that a genetic algorithm can yield very good optimisation results (Sorensen, et al., 2017), their performance is still very hard to predict. As such, a discrepancy between theory and experiments exists.

This discrepancy is thought to originate from decisions and assumptions that are made during the setup of an optimisation model for a problem. These include 1) the number and type (continuous or discrete) of design variables, 2) sets of design variables 3) the choice of starting solutions 4) number of design cycles 5) methods for handling infeasible solutions 6) number of independent runs performed and the design of the GA itself (Sorensen, et al., 2017)

As a high-level algorithm, the GA can be understood through the following steps, as shown in pseudo-code in Algorithm 1. First, a set $\mathcal{G}_{init}$ of initial solutions s_i where $i, \dots, N_{pop}$ are generated at random. For each solution s_i the objective function $F(s_i)$ is evaluated. As long as the termination condition T is not reached, a selection heuristic is applied to find a subset $\mathcal{G}_{parents}$ of the strongest solutions in which each solution is assigned a probability of reproduction $\mathrm{P_{r,i}}$.

Based on this reproduction probability, solutions are randomly selected to exchange genetic information using a crossover heuristic. Crossover is repeated until a set $\mathcal{G}_{offspring}$ of size N_{pop} is created.

Thereafter, random variation is introduced using a mutation heuristic and the resulting set becomes the $\mathcal{G}_{current}$.

This process repeats until the termination condition is reached at which the $\mathcal{G}_{current}$ will turn into the $\mathcal{G}_{final}$. The termination condition can involve a fixed number of iterations, a minimal increase in value of the previous generations, or any other criterion. As a result, the average solution value in terms of the objective function is likely to have increased over the generations. Important to note is that this concerns a stochastic search algorithm and therefore the algorithm might produce different outcomes for every run.

Algorithm 1: Canonical Genetic Algorithm [38]

```
repeat
    Step 1. Initialisation G_init ;
    repeat
        Step 2. Evaluation F(s_i) ∀ s_i ∈ G_current;
        repeat
            Step 3. Select Parents Solutions G_parents ;
            Step 4. Crossover Solutions G_offspring ;
        until Step 6.a N_offspring = N_pop;
        Step 5. Mutation of G_offspring;
    until Step 2.a Termination Condition is True;
until Step 2.b Termination Condition is True;
```

Each of the initialisation, selection, crossover or mutation steps can be viewed as an exchangeable component of the algorithm through which variations of the SGA can be constructed. While the SGA is applied in most cases, it can be shown that these standard heuristics do not always yield the best outcome (Wolpert & Macready, 1997). This implies that a GA can be designed for better or worse and thus requires us to understand the mechanisms through which it works.

Fogels claimed that genetic algorithms are not fundamentally different than any other optimisation technique as they traverse the solution space using a population in whatever direction that might lead to an optima (or peak). The only way genetic algorithms are different from other evolutionary algorithms is in the particular heuristics they use, specifically crossover and mutation.

Assuming that this is true, then we must thoroughly understand the roles and mechanisms of crossover and mutation. A common issue is that not enough well-defined ways of looking at genetic algorithm behaviour have been generated, inhibiting rigorous analysis and further empirical exploration (Mitchell, et al., 1991).

2.3.2. REPRESENTATION & TERMINOLOGY

In order for a GA to optimise a design problem each solution must be represented in such a way that the GA heuristics can modify the solution. Similar to nature, where our phenotypic expressions (e.g. eye colour) is represented by a gene in our DNA, the engineering design solution space must be encoded in so-called genotypes or chromosomes (these terms are used interchangeably). Consequently, GAs can work with a variety of problems without the GA heuristics needing to be problem-specific.

Figure 5 depicts a number of key concepts for the GA. First, the population is a set of solutions which are each represented by a chromosome. Each solution chromosome consists out of an array of genes. Each gene can have a value, which is called an allele.

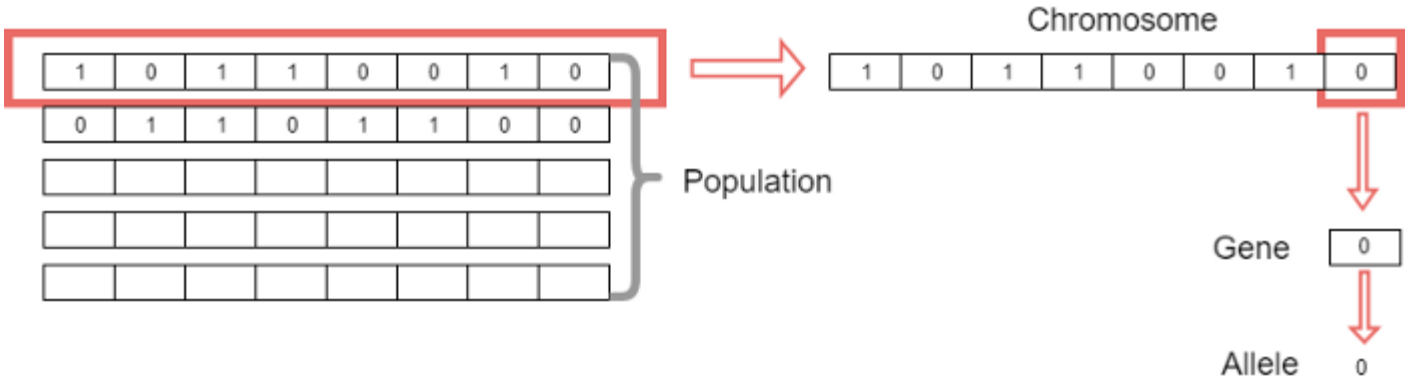


Figure 5 A schematic overview of a population of solutions encoded in binary chromosomes consisting of genes and alleles. Image downloaded from: (Shyalika, n.d.)

2.3.3. STANDARD CROSSOVER & MUTATION

The traditional view is that crossover is mainly responsible for exploitation of the genetic material of the existing population of solutions, while the role of mutation is of exploration of new genetic material by introducing new alleles into the population. However, the relative benefits of crossover or mutation are still debated amongst researchers (Senaratna, 2005).

Crossover (sometimes referred to as recombination) is the most complex of GA heuristics. Originally inspired by the genetic recombination during meiotic sexual reproduction, the general intention was to approximately imitate the effects of breeding in natural populations by combining the genetic information

of two (or more) parents in some manner. Do note that the specifics between biological recombination and GA crossover are very much different due to the many simplifications.

The original standard form of crossover as introduced by Holland (Holland, 1992) is one-point crossover, yet in most research and applications of GAs also two-point and uniform crossover are used (generally called n-point crossover) (Back, et al., 2000). Therefore these are viewed as the standard heuristics and are shown in Figure 6.

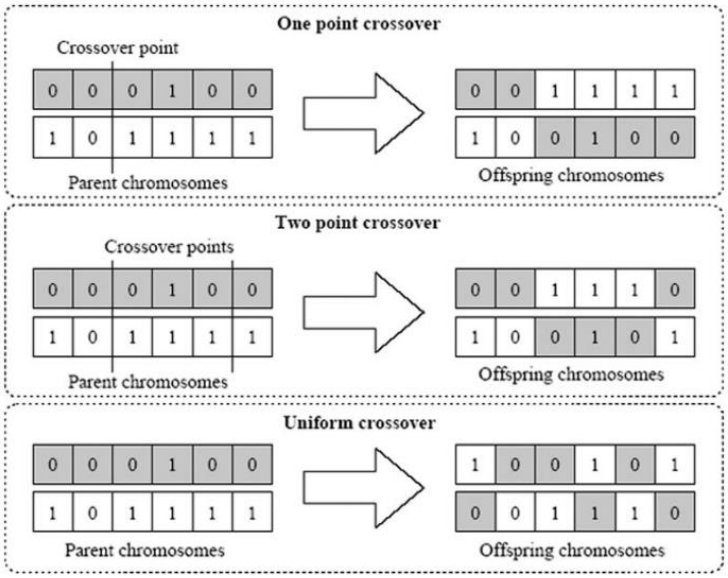


Figure 6 Standard crossover heuristics for binary chromosomes (obtained from (Goldberg & Sastry, n.d.))

These n-point crossovers take two parent chromosomes and based on n randomly chosen crossover points alternate the two parent chromosomes. As a result, two offspring chromosomes that inherit characteristics of their parent chromosomes are created. In the case of uniform crossover, a logical extensions of n-point crossover, each gene of either the parent alleles is chosen with equal probability.

The mutation heuristic has been the most elementary heuristic in the GA. Its primary function is to introduce new genetic material by randomly changing the genotypes of individuals in the population. It was inspired by the mutation observed in biological genetics where transcription errors in DNA led to minor changes in the chromosomes. Most research uses uniform mutation (real representation) or flip-mutation (binary representation) which is applied to each gene independently. As such, the number of genes changed in an individual is not fixed. While uniform mutation is perceived as the standard, many other mutation heuristics exist but their mechanisms are just as poorly understood (Soni & Kumar, n.d.).

2.3.4. GENETIC ALGORITHMS IN ENGINEERING DESIGN

In engineering design problems SGA's are generally applied in a black-box manner, meaning that without any knowledge of the mechanisms of optimisation or the nature of the solution space, the results are obtained through an arbitrary choice in the GA heuristics and fitting of parameters. One reason is that in contrast to purely mathematical analytical functions, the structure of the objective function (or fitness landscape in GA terms) in engineering design is very difficult to accurately describe due to imperfectness (noise, multimodality, high order dimensionality, deceptiveness) (Roy, et al., 2008). Another issue is that the implications of using specific GA heuristic combinations is not theoretically understood (Sorensen, et al., 2017).

Anecdotal evidence of the arbitrary choice of heuristics and parameters can be seen by sampling a number of engineering design studies which use GAs for optimisation, shown in Table 3. Each study makes use of standard GA components for crossover and mutation, while some studies don't have a mutation mechanisms

at all. Population sizes vary strongly, just like the crossover and mutation rates. Many employ additional strategies in order to cope with difficulties in optimisation specific to the problem definitions in those studies. Further details on the implementations of GAs in these studies can be found in the respective papers.

Table 3 Parameters of various engineering design studies with genetic algorithm ranging between 1993 - 2017. G = number of genes, P = permutations, - = not specified

Parameter	Lu et al. (2017) **(Lu, 2017)**	Fonseca et al. (1993) **(Fonseca & Fleming, 1993)**	Soremekun et al. (2001) **(Soremekun, et al., 2001)**	Cheng et al. (1998) **(Cheng & Li, 1998)**	Caraballo et al. (2017) **(Caraballo, et al., 2017)**
Object	Variable fuselage plate thickness	Gas turbine engine	Composite laminate	Truss structure	Butterfly valve disc
Dimensionality	$G = 5, 10$ $P = 1.1e^{12}$	-	$G = 14$ $P = 65536$	$G = 60, 120$	$G = 10$ $P = 12.9e^{9}$
Variable type	Binary	Binary	Integer	Binary	Real-valued
Constraint handling	Mapping function, data structuring	-	Data structuring	Penalty function	Tournament (Corne & Knowles, 2007)
Objectives	2	4	1	2	4
N	20, 32, 100	80	15-75	400	25
Pc	0.5	-	1.0	0.6	0.6
Pm	0.2, 0.2	-	0.01, 0.9	0.01	0.01
Selection	Tournament (3)	Rank	Linear Rank	Stochastic Remainder Selection (Goldberg & E., 1989)	Tournament (not specified)
Crossover	-	Two-Point	Single-Point	Uniform	Uniform
Mutation	Bit-flip	Bit-flip	Bit-flip & Swap	-	-
Additional strategies	Minimal weight objective as constraint	Progressive articulation of preferences DM, fitness sharing (Friedrich, et al., 2009)	Multiple- and variable elitist selection (Bäck & Hoffmeister, 1991)	Pareto-set filter (Marler & Arora, 2004)	Topology optimisation combined with NSGA-II (Deb, et al., 2002)

3. RESEARCH PLAN

Summarizing from the literature review, a number of knowledge gaps that prohibit the wider application GAs in engineering design were identified. First, the key functions of the GA are not clear. Based on what understanding do we evaluate the applicability of a GA design for a specific optimisation problem? How do we gain confidence in this? Literature did not contain any reasoning on this part. Second, it is not clear to what extent the roles of crossover or mutation heuristics in serving the functions of a GA are unique as literature showed no substantiation on why certain heuristics were used. Third, it is not clear what effect quantitative parameters such as problem granularity or heuristics-specific parameters such as the mutation rate have the optimisation process of a GA. Fourth, the objective functions are not evaluated in terms of whether they accurately capture the intention of the engineering designer, such that the optimisation leads to trivial or sub-optimal solutions.

Based on these knowledge gaps, the goals of this research became to increase understanding on 1) the mechanisms of the standard GA and 2) how to determine suitable objective functions.

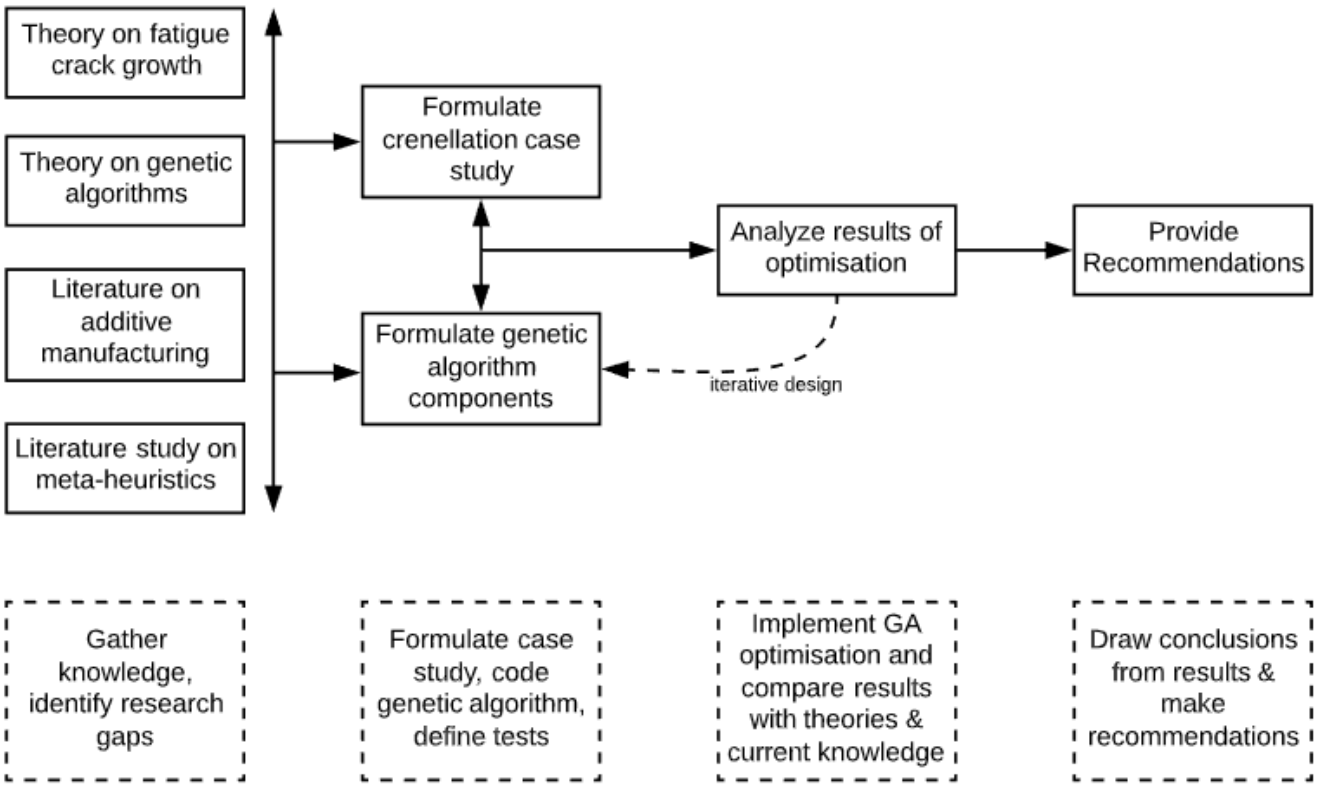


Figure 7 Research approach employing an iterative formulation of genetic algorithm components as more knowledge about the optimisation process is collected

To support the goals of this research an approach was determined, as schematically shown in Figure 7. First, the research efforts will be scoped by defining a set of research questions that will yield understanding in regards of the research goals, after which a suitable crenellation case study will be determined. A general framework in which the crenellation case study can be solved by a GA must be formulated along with a systematic way of testing these GA components. Certain choices on which heuristics and parameters are of most interest to investigate will be made in the Methods chapter. Finally, a method to investigate the performance of different GA design must be devised in order to draw some conclusions and provide recommendations for further research.

3.1. RESEARCH QUESTIONS & HYPOTHESES

In order to scope the research efforts, a number of primary and secondary research questions were determined.

1. **What is the relative importance of crossover and mutation heuristics in the optimisation process of a genetic algorithm designed for damage tolerance of flat panels?**

   1.1. Are there specific functions in the optimisation process of a GA which need to be fulfilled in this application?

      1.1.1. Our hypothesis is that specific GA functions can be defined in the optimisation process for damage tolerant flat panels

   1.2. How well do the standard crossover and mutation heuristics fulfil these functions within this context?

      1.2.1. Our hypothesis is that standard heuristics are not well suited for their intended functions in designed for damage tolerant flat panels

   1.3. Could a less random mutation heuristic increase the reliability of the GA in identifying optimal design solutions for damage tolerant flat panels?

      1.3.1. Our hypothesis is that the filtering of potential mutations in the mutation heuristic will improve the search results of a GA

   1.4. Under what conditions, if any, do we need both crossover and mutation heuristics in a GA designed for identifying optimally damage tolerant flat panels?

      1.4.1. Our hypothesis is that crossover is not a necessary component in GA optimisation for damage tolerant flat panels

2. **To what extent do quantitative parameters of the GA influence the optimisation process when designing for damage tolerant flat panels?**

   2.1. What correlations are there between the population size and the reliability of a GA for varying values of the population size?

      2.1.1. Our hypothesis is that a larger population size leads to higher reliability of the GA in finding optimal results

   2.2. What correlations are there between the mutation rate and the reliability of a GA for varying values of the mutation rate?

      2.2.1. Our hypothesis is that a higher mutation rate will not always lead to higher reliability of the GA in finding optimal results

3. **Can a simple objective function be formulated that captures the intentions of a designer a priori?**

      3.1.1. Our hypotheses is that simple objective functions will a priori lead to trivial solutions which the designer did not intend to achieve

      3.1.2. Simplified models may be sufficient to capture the essential behaviour of a genetic algorithm

4. METHODS

To answer the research questions and study the behaviour of GAs in an engineering design application, a simple GA optimisation routine was developed for a relevant engineering design problem – the optimisation of thickness distribution for a crenellated fuselage skin panel. The basis for this optimisation is the damage tolerance behaviour of the panel in the presence of a fatigue crack. The choice for this design problem was made due to the simplicity of defining the design optimisation in a GA framework, the availability of analytical models that describe the damage tolerance behaviour, and the potential to critically evaluate the resulting optimized designs.

This chapter will outline the details of the GA optimisation framework used in this study, how the design problem was formulated and integrated into this framework and how the performance of the GAs was systematically studied.

4.1. OVERALL GA FRAMEWORK

Through an iterative process the routine depicted in Figure 8 was created for the optimisation of thickness distribution for a crenellated plate.

The overarching optimisation routine starts with the definition of a mathematical formulation (block 1) of the design variables and the objective of the design optimisation. Its primary function is to ensure an encoding of the design problem such that GA heuristics are applicable and the analytical fatigue model can be used to accurately evaluate damage tolerance of each solution. Furthermore, it incorporates a notion of feasible solutions through equality and inequality constraints.

The second block in the routine concerns the genetic algorithm (block 2) and it is configured using a number of GA parameters and heuristics. Its primary function is to vary solutions using a set of heuristics in order to search the space of possible design solutions. The GA heuristics used can be either standard or custom to this research, which will be expanded upon in Section 4.3.

Following the application of heuristics to a set of solutions, a termination condition is checked, and if not triggered, the set of design solutions is evaluated using a fatigue crack growth model (block 3) for crenellated plates. Its primary function is to ensure an accurate estimation of the damage tolerance for a given

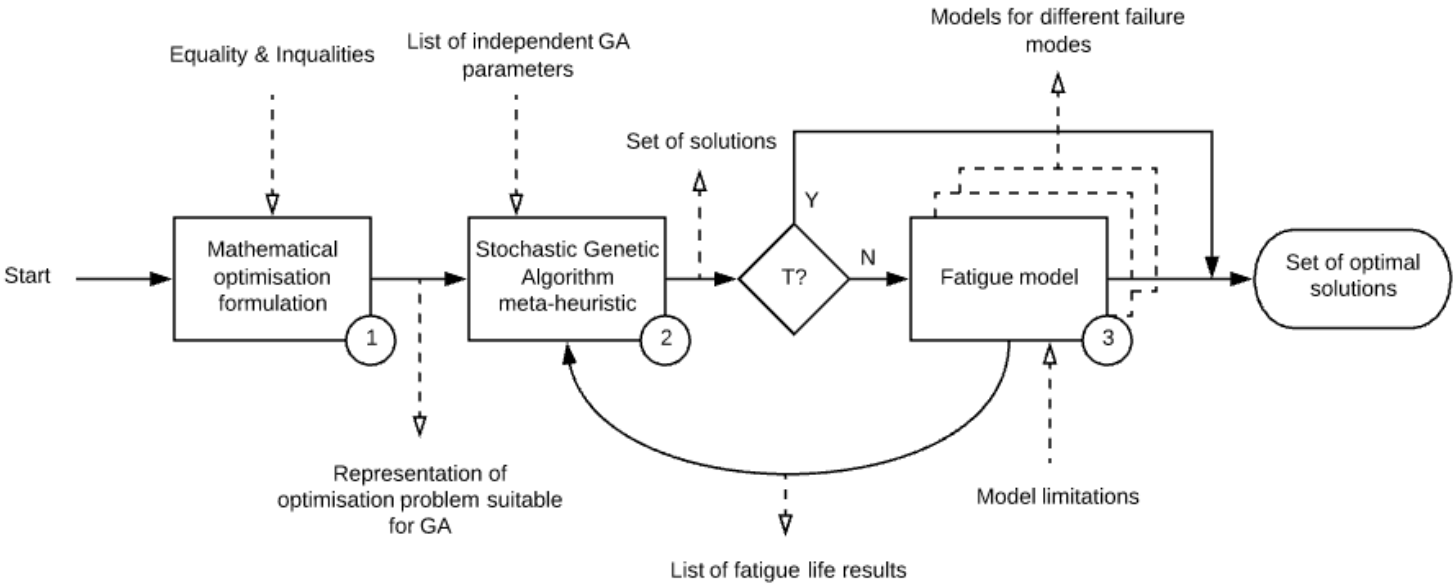


Figure 8 Overarching routine for a design optimisation run including 1) problem formulation, 2) genetic algorithm and 3) fatigue crack growth model for crenellated plates

crenellated plate design solution.

The overarching model can be extended with a number of models which evaluate different failure modes. Fatigue crack growth was chosen due to the availability of an analytical model which greatly reduces the computation time of a design optimisation run compared to FEM models, an issue raised by Lu et al (Lu, et al., 2016). This gives us the ability to study the genetic algorithm mechanisms in more detail, the main interest of this research.

The fatigue crack growth results are passed back to the GA after which it applies heuristics for recombining and mutating the set of solutions based on these results. This cycle continues until the termination condition is triggered and at that point the design optimisation run is completed.

The termination condition for this study was defined as 20 iterations between the genetic algorithm and the fatigue model, as it was empirically determined to reveal most of the optimisation behaviour while minimizing unnecessary computation. The final output of the overarching model is a set of more optimal solutions. As this process is stochastic, each experiment was run 10 times in order to capture the average behaviour of the optimisation process. This number was iteratively determined by looking how much the end-state of the optimisation would change for every additional run. In order to compare results amongst different experiments, the randomness was seeded such that for each experiment, the results would be reproducible if started with the same seed, rather than be different for every time an experiment would be run.

4.2. DESIGN PROBLEM FORMULATION OF CRENELLATION

The following section describes the detailed formulation of the crenellation problem (block 1) in the optimisation routine.

Consider a plate of width W with a systemically varying thickness described by a function $t_{n_i}(x)$ where x is the distance along the plate width where $0 < x < W$. Depending on the granularity γ of the problem formulation, the plate width is divided into a number of partitions n_i of equal width Δx where $1 < i < \gamma$ and $\Delta x = \frac{W}{\gamma}$. Each partition can have a thickness t_{n_i} which is independent from the other partition thicknesses.

A crack of length a_0 is introduced and the crenellated plate is repeatedly loaded with a far-field stress S_{max}. Due to this loading, the crack grows incrementally with Δa for each load cycle ΔN. The loading continues until the crack reaches length a_{max}. At that point, the crenellated plate will have sustained a number of load cycles N_{life}.

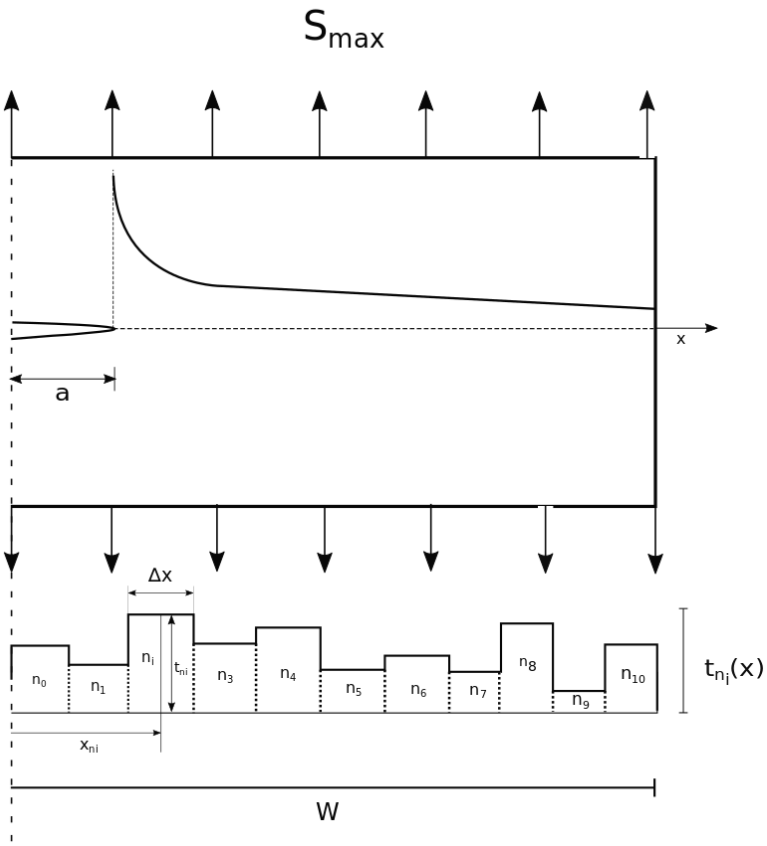


Figure 9 Design problem formulation of a crenellated plate loaded by a far-field stress

The optimisation problem can be defined as finding a combination of design variables s which maximizes an objective function $F(s)$, such that

$$\hat{s} = \max_{s \in \Omega} F(s)$$

Where the vector $\hat{s}$ is the global maximum.

4.2.1. DESIGN REPRESENTATION

The optimisation problem of crenellation patterns was formulated as a combinatorial optimisation problem using an indirect integer representation. An indirect representation, consisting of both a genotype-phenotype and phenotype-fitness mapping, was chosen because it improves the reproducibility and generalizability of the insights in this research. Due to this representation, the standard and problem-independent crossover and mutation heuristics can be further developed and easily tested on other problems in literature. Furthermore, it makes it possible to control the size of the genotypic solution space, and thus problem complexity, through the granularity parameter γ. This control over complexity made it easy to investigate how these GA heuristics perform under ranges of complexity.

As such, the fitness function is decomposed into two parts,

$$F_g(s): \Omega_g \rightarrow \Omega_p$$

$$F_p\left(t_{s_i}(x)\right): \Omega_g \rightarrow \mathbb{R}$$

Where $F = F_p{\circ}F_g = F_p\left(F_g(s)\right)$. The genotype-phenotype mapping F_g is determined by the representation. F_p represents the fitness function which assigns a fitness value to any solution $s \in \Omega_p$. The GA heuristics are applied to the solutions in Ω_g.

Integer representation was chosen for similar reasons, as the discreteness largely reduces the number of possible thickness levels, as compared to a floating point and continuous representation. Contrary to a binary ($\boldsymbol{\chi = 2}$) representation, an integer ($\boldsymbol{\chi - ary}$) representation does offer more flexibility to define the exact granularity one wishes to optimise for. It was expected that small variations in thickness levels would not contribute to better insights on the mechanisms of GA heuristics [ref], thus the problem formulation was not investigated and limited to 3 thickness levels such that $\boldsymbol{\chi = 3.}$

As such, the genotypic solution space can be described as,

$$\Omega_g = \{0,1,2\}^{\gamma}$$

With the length of the vector equal to γ and the size of the solution space $\left|\Omega_g\right| = \chi^{\gamma}$.

Careful consideration was given to further reduce the biases of representational mappings. First, the genotype-phenotype mapping was chosen uniformly redundant, meaning that each phenotype is represented by the same number of genotypes, namely a 1-to-1 relation in this case. As a result, no neutral sets of solutions exist and therefore no neutral mutation of crossover exists.

Moreover, a high locality representation was defined to ensure that the difficulty of the problem is not worsened due to a mapping which introduces additional randomness (Rothlauf, 2006). Third, a uniform scaling representation was defined since non-uniformly scaled representations have shown to modify the

dynamics of GA search in not well understood ways (Rothlauf, 2006). As the subjects of this research are the GA heuristics & parameters, it was chosen to exclude this influence. Figure 10 shows the genotypic and phenotypic representation of the design problem for $\gamma = 6$.

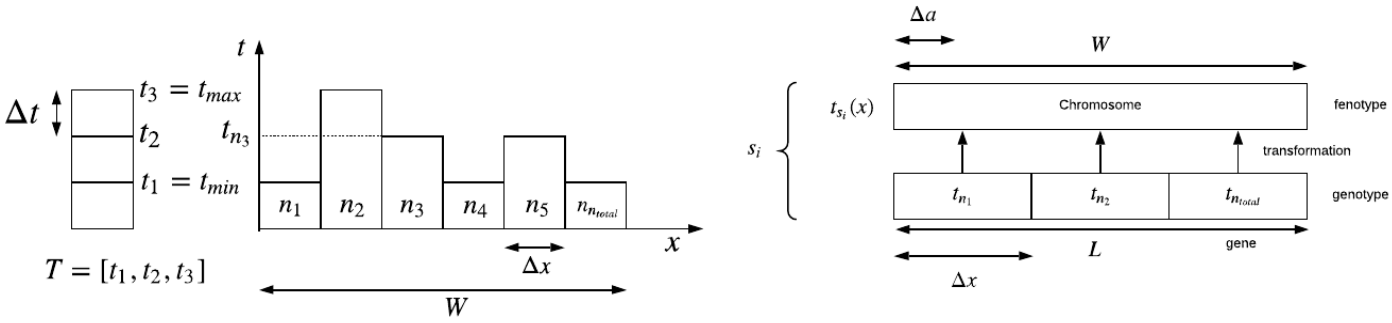


Figure 10 a) crenellation pattern (γ=6) and b) encoding of crenellation pattern in a genotype and phenotype

4.2.2. OBJECTIVE FUNCTIONS

The goal is to optimize the fatigue life N_{life} of the plate through the distribution of thickness over the partitions, while minimizing the weight of the plate. In order to evaluate the value of a given crenellated plate design, an objective function that best captures the designers intent must be determined in mathematical terms. This objective function will yield a score for each design solution which the genetic algorithm can use as a signal for the optimisation of the design problem. A well-known limitation of the SGA is that it can only work with a single numeric value. Therefore, multiple objectives must be reduced to a single value.

In this research, three objective functions are investigated. The first objective function $F_1(s)$ only considers the fatigue life N of the crenellated plate. As one might imagine, a designer might initially only be interested in finding solutions with the best performance on fatigue life. This objective function captures just that intention.

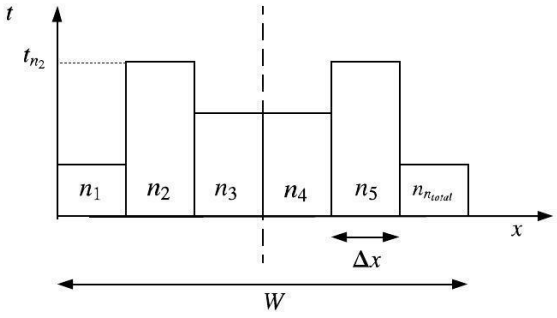
The second objective function considers the fatigue life N over the cross-sectional surface area A of the plate. In this case, the designer has understood that he wants to optimise for fatigue life per unit of weight and has taken the cross-sectional surface as a logical expression for the plate weight.

A third objective function contains the addition of the material constant m as a power function on the cross-sectional surface area A of the plate. As known in the Paris equation, the material constant has a strong influence on the crack growth rate in a plate. At this point, the designer has realized that the fatigue crack growth model contains an expression with this material constant and has included it in the objective function.

Table 4 Overview of the objective functions investigated in this research

$$F_1(s) = N \qquad F_2(s) = \frac{N}{A} \qquad F_3(s) = \frac{N}{A^m}$$

4.2.3. DESIGN CONSTRAINTS



One of the challenges when defining the solutions space for GAs is ensuring that the produced offspring solutions remain feasible. In this study, a symmetry design constraint was applied to the genotypic solution space to ensure that the fatigue life of design solutions would not be dependent on the crack starting location on either side of the plate. It is assumed that cracks start at either of those locations since these are the locations which have high stress concentrations due to stringers. Every design solution complying with this design constraint is part of the feasible genotypic solution space $\mathcal{F}_g$.

$$F_p\left(t_{s_i}(x)\right) : \mathcal{F}_g \rightarrow \mathbb{R}$$

As a result of this constraint, the number of partitions with variable thickness is divided into half. Second, the allowable thickness levels for each gene were constrained to 3 distinct levels, namely the allele translation from integers to thickness was fixed 0:2.0, 1:3.0, 2:4.0mm

Table 5 Summary of the default problem definition settings unless stated otherwise in the experiment setup

W (mm)	a_0(mm)	a_{max}(mm)	Δa (mm)	t_{min} (mm)	t_{max}(mm)	Δt (mm)	γ (-)	χ (-)
150	37	145	1	2	4	1	30	3

4.2.4. MATERIAL PROPERTIES & LOADING CONDITIONS

Table 6 provides an overview of the material properties used in this research in comparison with the reference study on crenellation. For the loading a stress of $S_{max} = 22000\ MPa$ was employed.

Table 6 Summary table of material properties used in this research.

Name	C [-]	m [-]	E-mod [MPa]	Poisson [-]	Source
AL2024-T3	1.86e-11	4.05	72500	0.33	Efatigue.com
AL2139-T8	2.74e-7	2.6	72400	0.33	Reference study (Uz, et al., 2009) (Lu, et al., 2016)

4.2.5. ASSUMPTIONS & LIMITATIONS

The problem definition is not representative of an (aerospace) design challenge. The design problem is highly simplified for the purpose of investigating a less complex interaction between the GA and the design problem. As such it is sufficient to gain a better understanding of the GA, which is the primary goal of this research.

A number of assumptions & limitations underpin the simplification:

- The design problem is limited to a single bay while not having stringers on both sides of the fuselage
- It assumes an initial crack starting point on the left side of the plate, originating from a stress concentration at the base of a hypothetical stringer.
- The model assumes a crack growth from an initial crack length to a final crack length in order to ensure that the crack growth can be described by the fatigue crack growth model within the linear Paris range
- The problem assumes a straight crack front where no turning of the crack is permitted. With crenellation patterns of identical material and relatively small variations of thickness, it is assumed that the lower and upper crack front progress equally fast thus creating a straight crack front.
- The model assumes a perfect variation in fatigue stress with no additional, secondary forces due to temperature, plastic deformation or impact
- The problem definition assumes that there are no defects in the crenellation shapes
- Also, the Paris equation constants C and m are known to be affected in some degree by specimen thickness, which is not taken into account in the model (Schijve, 2009)

4.3. ANALYTICAL FATIGUE MODEL FOR CRENELLATED PLATES

To evaluate the value of feasible crenellation design solutions in terms of fatigue damage tolerance, a model which can predict the fatigue life of a given crenellated plate was needed. For this, the analytical model for bonded stiffeners by Rans et al (Rans, et al., 2015) was adapted to be suitable for crenellated plates of a single material.

4.3.1. MODEL

The model summarized by Equation 10 aims to calculate the fatigue life N for a fatigue crack growing from an initial crack length a_0 to a final crack length of a_{max} over a crenellated plate of shape $t_{s_i}(x)$, such that

$$N = \int_{a_0}^{a_{max}} \frac{\Delta a}{C * \left(\sigma_{eff}\sqrt{\pi a}\right)^m} = \int_{a_0}^{a_{max}} \frac{\Delta a}{C * \left(\frac{S_{max}\sqrt{\pi a}}{2 * \int_a^W \frac{\frac{t_{s_i}(x)}{\sqrt{1-\left(\frac{a}{x}\right)^2}}}{} dx} \right)^m} \quad (10)$$

For more details on the intermediate steps of creating the model the reader is referred to either Appendix B of this research or the paper on bonded stiffeners (Rans, et al., 2015).

The primary function of the analytical model is to capture the damage tolerance behaviour with sufficient accuracy to allow for the study of GA heuristics, the main object of investigation of this research. To validate the suitability of this model for that purpose, its predictions were compared with experimental data from the reference study by Uz et al (Uz, et al., 2009) of which the outcomes will be shown in the Results chapter.

Another important motivation to choose this analytical model is that it takes a short time to evaluate as compared to FEM alternatives, even for highly granular definitions with γ = 150, which enables the investigation of how the GA interacts with these problem definitions without being constrained by the evaluation time, which was mentioned as an important constraint in the reference study (Lu, et al., 2016). This also enables the investigation of heuristics that require more frequent evaluations.

This model introduces additional assumptions & limitations as compared to those mentioned in the problem definition section. First, the model only accounts for the stiffness ahead of the crack tip when calculating the stress at the crack tip for each increment, not behind it. Second, the Paris equation constants C and m are known to be affected in some degree by specimen thickness, which is not taken into account in the model (Schijve, 2009). Third, the model assumes a perfect variation in fatigue stress with no additional, secondary forces due to temperature, plastic deformation or impact.

4.3.2. VERIFICATION OF FATIGUE MODEL

The analytical model for fatigue crack growth in crenellated plates as introduced in the previous chapter was verified with experimental data from Uz et al (Uz, et al., 2009). Figure 13 and 14 show the results of the model verification with a double bay flat plate containing a single large thickness increase, as well as a single bay crenellated plate. These were patterns were chosen due to the availability of experimental data. The material properties and loading conditions were summarized in the methods chapter. A number of discrepancies between the analytical fatigue model prediction and experimental data can be found.

First, the model underestimates crack growth rates for all crack lengths where 37 < a < 247 mm in a flat plate.

Second, the model overestimates the crack retardation effect towards larger thickness steps (15mm). Both in Figure 13 (a = 143) and Figure 14 (a = 143) the model crack growth rate prediction rapidly decreases.

Third, the model underestimates the crack retardation effect towards smaller thickness steps (+- 3mm), especially in the longer period working towards the step increase in thickness. For instance, in Figure 14 where 37 < a < 55 mm and 100 < a < 120 mm.

Fourth, the model underestimates the crack acceleration after a smaller and larger thickness decrease (+- 3mm and 15mm). In Figure 13 we can see that after the crack has progressed through the 15mm stringer, the model underestimates the crack growth rates consistently as the slope of the experimental data da/dN graph is much steeper than the slope of the model prediction. In Figure 14 we can see a similar prediction error for smaller thickness decreases, as the initial crack growth rate of the model following a thickness decrease (a = 37mm) is too low, as well as the peak in crack growth rate at the second thickness decrease (a = 95 mm) is missing from the analytical model prediction. On the contrary, the FEM model prediction by Huber et al highly overestimates this peak.

A likely reason for this underprediction is that the analytical model does not take into account the additional stress at the crack tip due to broken material in the wake of the crack. As such, at a thickness decrease, the crack growth rate should rise strongly since the relatively large area of material in close proximity to the crack tip carries a large part of the load (Rans, et al., 2015).

Fifth, model does accurately predict the crack growth rates on the thicker crenellated sections, such as in Figure 14 where 60 < a < 95 mm and 120 < a < 140 mm.

Last, the model does not take into account load history effects, which can effectively reduce the crack growth rate in thinner plates due to the creation of plastic zones around the crack tip (Schijve, 2009).

Undoubtably, this analytical model does not capture the crack growth behaviour of crenellated plates very well. Nonetheless, it does capture some important characteristics with which the genetic algorithm will be able to differentiate between design solutions. As our research questions are more focussed on understanding the behaviour of genetic algorithms, the current engineering model was considered sufficient for the application on this case study.

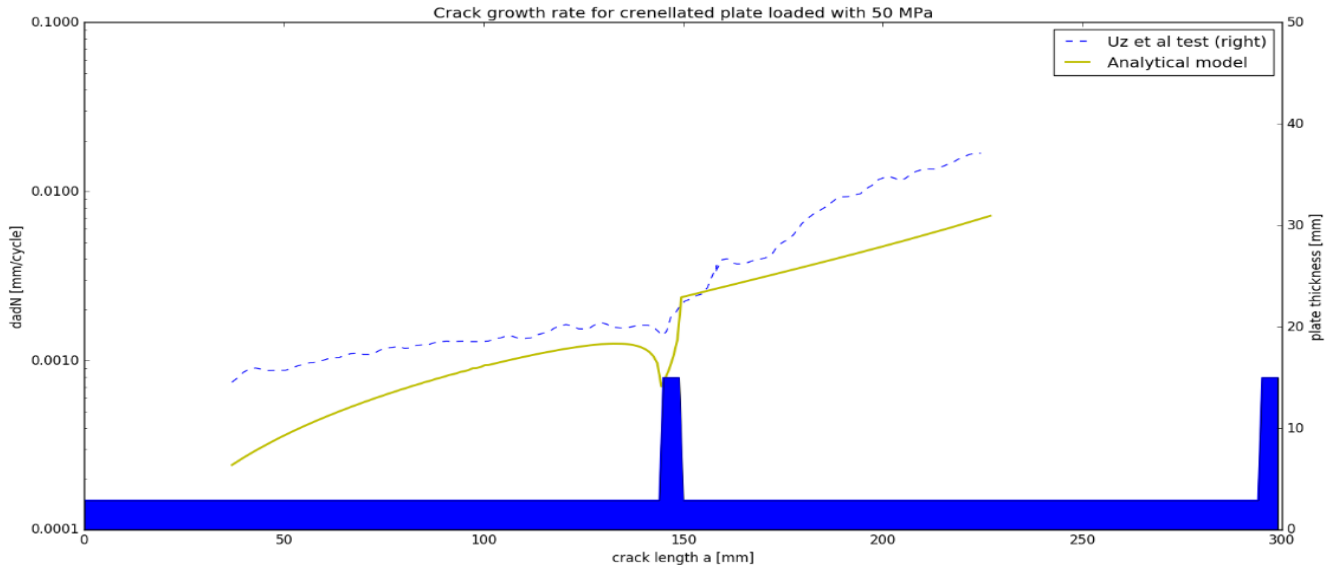


Figure 12 Verification of the analytical FCP model for a double bay flat plate configuration (Uz, et al., 2009)

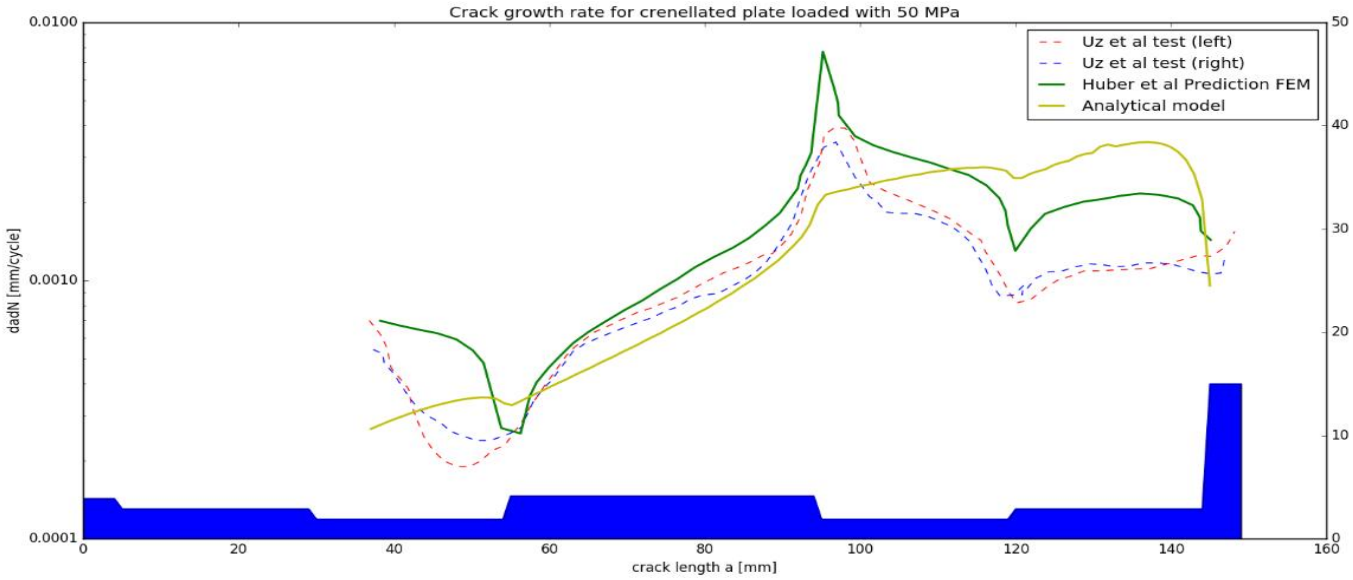


Figure 12 Verification of analytical model for a single large thickness deviation (Uz, et al., 2009)

4.4. BRUTE FORCE OPTIMISATION

Provided the (simple) analytical model and the crenellation design problem definition, it was possible to determine the true optimal solution through a complete enumeration and evaluation of all possible solutions in a reasonable amount of time for granularities up to γ = 15, $\chi = 3$.

The primary goal of this research is the better understand how the GA heuristics interact with the problem definition of optimisation a crenellated fuselage plate.

In order to investigate the behaviour, a transparent method was necessary. Transparency means understanding if what the GA was doing with the alleles is something preferable or not. To know this, the end-state of a given allele should be known.

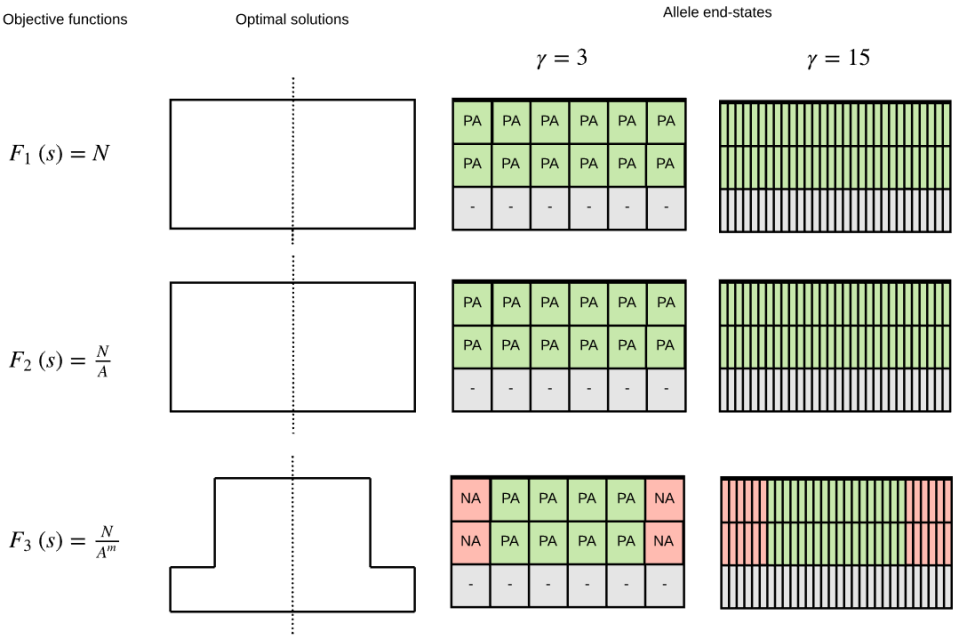


Figure 13 Schematic overview of the optimal solutions per objective function grouped by granularity

For each objective function for a course granularity of γ = 3 and γ = 15, we can approximate the optimal solutions for two levels of granularity through a brute-force method of enumeration of all possible solutions. Alleles that are part of the optimal solution, we call Positive Alleles (PA, green) and those that are not Negative Alleles (NA, red). These are custom definitions for this research and their meaning will further be expanded upon in Section 4.5 when introducing the custom heuristics. **Error! Reference source not found.** shows the optimal results for each o bjective function.

4.5. HEURISTICS

In order to influence the optimisation of a GA, several standard and custom heuristics were investigated. The custom heuristics for this research focused on mutation, while for crossover only standard heuristics were considered, as the broad definition for mutation, that of a random change to a genotype, allowed for a much wider design freedom for custom heuristics to search the solution space.

This section first introduces a number of custom definitions and measures used in this research, as these provide the basis of understanding the custom mutation heuristics introduced thereafter. Furthermore, a number of custom initialisation heuristics are introduced. The section concludes with a summary of default values for hyperparameters in the optimisation.

4.5.1. CUSTOM DEFINITIONS & MEASURES

POPULATION RELATIVE ALLLE STRENGTH

The measure *Population Relative Allele Strength* $PRAS_j$, was created to demonstrate how relatively important in terms of population fitness a given allele j is in the population and how this develops over the

generations. Given that the number of alleles in a population ranges between $0 < j < \gamma \cdot (\chi - 1)$, PRAS is calculated as follows,

$$PRAS_{j,eff} = \frac{\sum_i F(s_i)\theta_{i,j}}{F_{pop}} \qquad (1)$$

Where F_{pop} is the total fitness of the population, $\theta_{i,j}$ a binary value which tells whether a certain solution s_i contains the allele j through a delta function where $\theta_{i,j} \in \{0,1\}$ and $0 \leq PRAS_j \leq 1$. The delta function is evaluated as follows,

$$\theta_{i,j} = \delta(A_j - a_{i,j}) \qquad (2)$$

Which only returns a binary value based on the delta of two values, such that $\delta(0) = 1$ and else $\delta = 0$. We have positive alleles where $A_j = 1$ and negative alleles where $A_j = 0$ as determined by the objective function and depicted in Figure 14. The variable $a_{i,j}$ resembles whether the solution i contains the given allele j ($a_{i,j} = 1$), or not ($a_{i,j} = 0$). The difference between the optimal allele value A_j and solution allele value $a_{i,j}$ is placed in the delta function, ensuring that both positive and negative alleles optimal states converge to $PRAS = 1$. In short, when a solution does not contain the optimal state for a given allele at the time of evaluation, its fitness is not attributed to the PRAS of the allele being evaluated.

Either when the optimal end-states of alleles are not known, or for graphs on a more granular level, the non-effective PRAS can be calculated, which means that A_j is always equal to 0, as we do not know the optimal end-state. According to Eq 1 and 2, NAs then navigate to PRAS = 0 and positive ones to PRAS = 1, which also allows for better readability when more graphs are presented, such as in Figure 14.

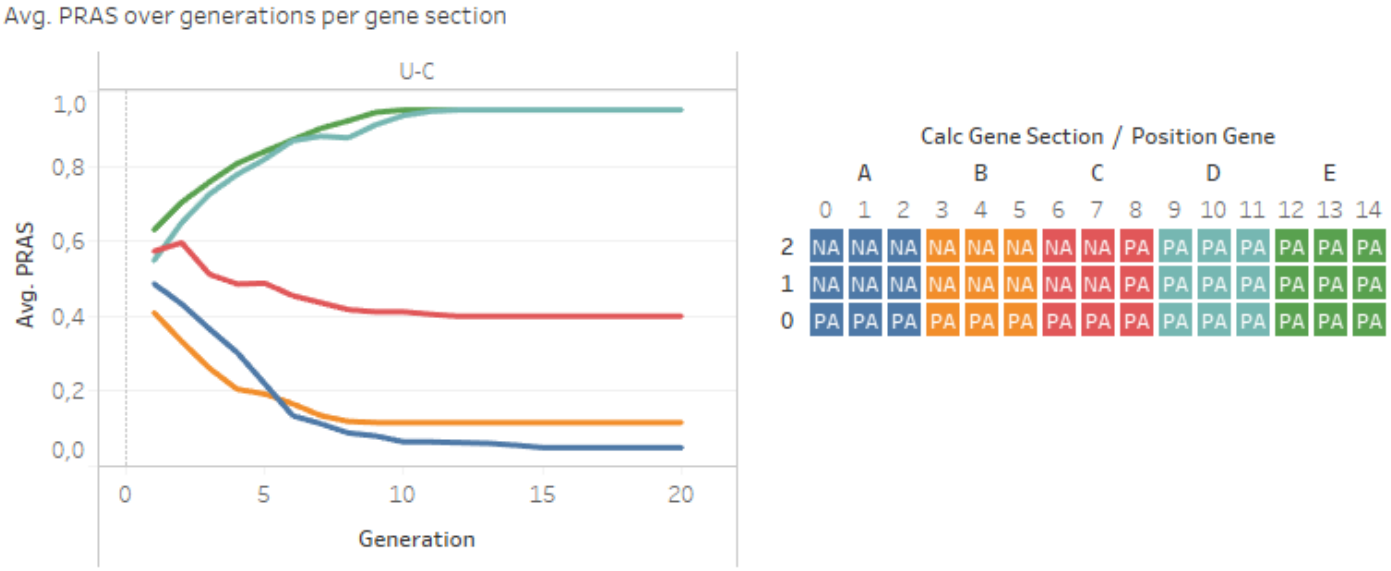


Figure 14 This figure shows an example of the relative strength per gene section and allele (right, γ=15, F=F₃) for a single GA run only using uniform crossover.

Important to note is that PRAS is not equivalent to the diversity of alleles in a given population, nor does it explicitly represent the fitness of a single allele. It represents the relative portion of population fitness that an allele is part of and thus might implicitly provide us information on the strength of the concerned allele when normalized based on its frequency in the population. The base layer of alleles is not included in the PRAS calculations, as they are always present in every solution and therefore will have PRAS = 1.

As an example, Figure 11 shows the average PRAS per gene section for 10 trials of an experiment with U-C, where γ = 15, $\chi = 3$. As such, the graph shows how the contribution to fitness an allele has over time. A population is fully optimised when all positive alleles have reached the upper range (PRAS=1) and all the negative alleles the lower range (PRAS=0). In the Figure we can see that the gene sections D and E, containing only PAs, have been fully optimised at PRAS = 1, while gene sections A and B, containing only NAs, are not fully optimised towards PRAS = 0.

Table 7 Overview of allele naming conventions based on end-state PRAS values

	$\boldsymbol{PRAS_{j,eff} = 1}$ **("ubiquitous")**	$\boldsymbol{PRAS_{j,eff} = 0}$ **("extinct")**
PA	Fully present in all solutions	Not present in any solution
NA	Not present in any solution	Fully present in all solutions

At a given generation, the population relative allele strength over frequency (PRASF) can be calculated as,

$$PRASF_j = \frac{\sum_i F(s_i)\theta_{i,j}}{F_{pop}} * \frac{1}{F_j} \qquad (4)$$

where F is the frequency of occurrence of allele j. The addition of the population frequency of an allele is a method for normalization to allow for a proper comparison between PRAS values.

GA FUNCTIONS, EVENTS AND RATES

Given the concept of allele end-states, a number of GA functions which the GA should be designed for can be determined. After all, each PA must obtain a PRAS of 1, while each NA must obtain a PRAS of 0. The hypothesis is that a good GA design should maximize the performance of the GA on these functions. Figure 15 provides a visual explanation of the 8 different GA functions based on whether an allele has a positive or negative end-state.

For a PA, the GA function should be reintroduce extinct PAs and subsequently progress them (no. 1 and 2, green), while avoiding the elimination and suppression of those PAs (no. 3 and 4). Inversely, for a NA, the GA function should be to avoid reintroducing extinct NAs and progressing them (no. 5 and 6). It should, however, eliminate ubiquitous NAs and suppress them up unto extinction from the population (no. 7 and 8).

On a practical note, the GA function events are registered at each generation for each allele during the optimisation. If the allele decreased in PRAS, a suppression event is registered. Alternatively, if its PRAS increased, then a progression event is registered. If an allele becomes ubiquitous due to a PRAS of 1, a

ubiquity event is registered. In similar fashion an extinction event is registered when an allele has a PRAS of 0.

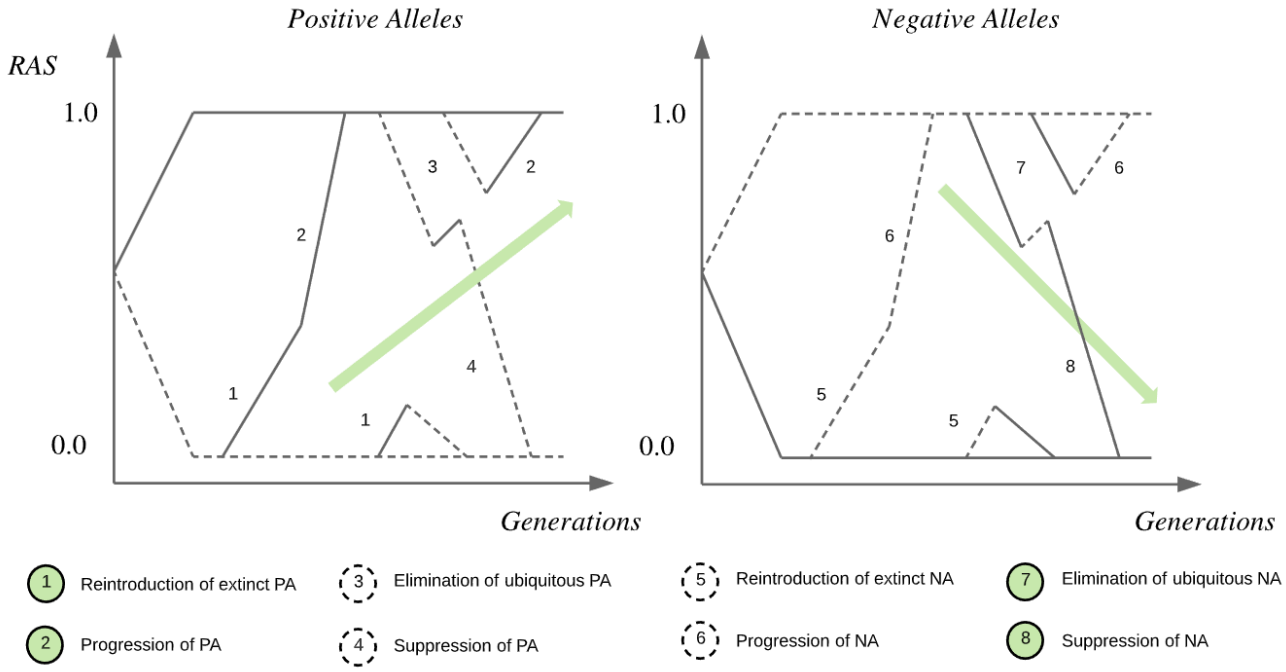


Figure 15 Schematic explanation of GA functions of introduction, progression, suppression and elimination in terms of PRAS per allele end-state

Elimination events are registered when an ubiquitous allele is removed from a solution, leading to a PRAS < 1. Vice versa for (re)introduction events, where PRAS becomes larger than 0.

GA function rates resemble the average change in PRAS for a single event and are calculated for the different GA functions, introduction, progression, suppression and elimination. The effective function rates are calculated by dividing the total the PRAS by the associated events for the desired functions (no. 1, 2, 7 and 8, green). Thus, the effective progression rate can be calculated as,

$$PR_{eff} = \frac{\Delta PRAS}{\Delta E} = \frac{\sum PRAS_{PA,progressed} + \sum PRAS_{NA,suppressed}}{\sum E_{PA,progressed} + \sum E_{NA,suppressed}}$$

Where E represents an event. This measure shows how much on average the events contributed to the GA functions and tells how effective the heuristic design was in choosing which solutions to alter using which alleles. A more effective GA design is expected to have higher effective GA function rates, rather than simply have more events associated with the desired GA functions.

Finally, an aggregate GA function score can used for quickly comparing GAs in order to understand to what degree they perform what we intended. This function does not fulfil a complete evaluation on the quality of such a GA and should be interpreted with caution. It can be calculated as follows,

$$GA_{score} = \Big((PR_{eff} - SR_{eff}) * (PE - SE)\Big) + \Big((IR_{eff} - ER_{eff}) * (IE - EE)\Big) * (PRAS\ end\ state - PRAS\ initial\ state)$$

Where PR = progression rate, SR = suppression rate, IR = introduction rate, ER = elimination rate, PE = progression events, EE = elimination events, SE = suppression events and IE = introduction events.

The extinction and ubiquity are not defined in the GA score because they are considered to be a resultant of the mechanisms that are designed for increasing the effective progression and/or suppression rate, while introduction and elimination are functions that require a different mechanism. As such, the GA functions score should not double count the contribution of such progression or suppression mechanisms.

4.5.2. CUSTOM MUTATION HEURISTICS

The following section will describe the detailed mechanisms of the custom mutation heuristics used in this research. The intention was to avoid particular naming as much as possible in order to improve the reproducibility of this research, however some naming specific to this research was unavoidable. Table 8 provides an overview of the 5 custom mutation heuristics, their components and 3 associated meta concepts. These meta-concepts were distilled in order to make comparison with other research more convenient. To the best of our knowledge, research that has applied an identical meta concept are listed in the table.

Table 8 Custom mutation heuristic components grouped per mutation heuristic with references to meta concepts and literature if available

Custom mutation components	**Relative Strength (RS)**	**Adaptive allele (AA)**	**Solution Memory (M)**	**Strong solutions (SS)**	**Enumeration of Ubiquitous and Extinct alleles (EUE)**
Meta concept	***PRAS Signalling***	***Mutation Filtering***	***Mutation Filtering***	***Selection bias***	***Selection bias***
Literature references (if any)		***Glover et al,*** (Forrest & Mitchell, 1993)			
1. RS-M	X				
2. RSAA-M	X	X			
3. RSM-M	X		X		
4. RSAASS-M	X	X		X	
5. RSAASSEUE-M	X	X		X	X

The 3 meta concepts are 1) PRAS signaling, 2) mutation filtering and 3) selection bias. A general introduction is provided below, where a more detailed description is provided in each of the custom mutation heuristics.

PRAS signaling

The standard GA applies a uniformly random mutation to determine which gene to alter to which allele. This meta concept changes that notion by removing the pure randomness with a signal using PRAS. Rather than uniformly random choosing which gene and allele the mutate, the PRAS values of each allele influence the probability that an allele is mutated towards for a solution that does not yet have this allele.

Mutation filtering

This concept applies a filter to applying the proposed mutations by the GA. The standard GA does not have this, it applies every mutation which is "proposed" by the algorithm. In this research, the filtering is done based on the incremental fitness that such a mutation would provide for the solution. Based on that incremental fitness, a rule can determine whether a proposed mutation should be applied to the solution or not.

Selection bias

The standard GA has a selection bias when it comes to crossover, as solutions with a higher fitness generally have a higher chance of being selected as a parent in most selection heuristics. However, mutation heuristics do not have such a notion in the SGA. The selection bias in this research applies to mutation heuristics in the sense that a solution with a high relative fitness in the population has a higher change to be mutated.

Mutation 1. | Relative strength (RS-M)

Concepts included are PRAS Signalling

Relative strength mutation was designed to have a bias towards the top 50% of alleles with the highest ratio of relative strength over frequency in the current population. The alleles are sorted in descending order based on the PRASF of the alleles. If each of these alleles are already present on the given solution, a zero frequency allele is chosen uniformly random and added to the solution. If there are no zero frequency alleles, mutation does not take place.

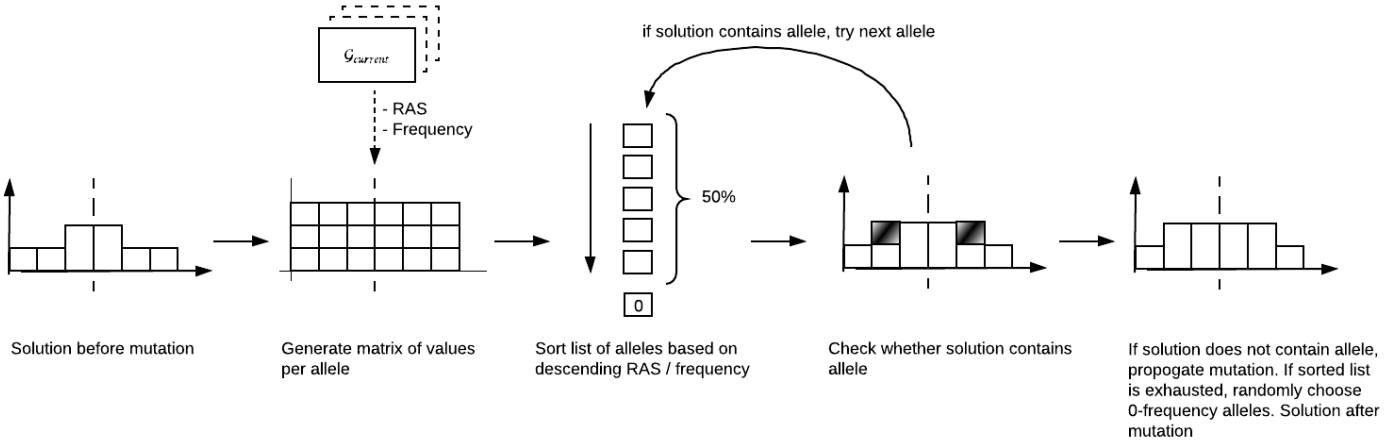


Figure 16 Schematic overview of relative strength mutation (RS-M) mutation

As such, this mutation introduces a more volatile environment where alleles which after introduction improve the fitness of their solution significantly have a much higher probability to be introduced on other solutions in a following mutation event. If there are many allele with the same high value of PRAS, the heuristic loops through the population matrix from top left to right going from top row to bottom.

Mutation 2. | Relative strength adaptive allele (RSAA-M)

Concepts included are PRAS signalling and mutation filtering.

Inspired by Lamarckian evolution (Whitley, et al., 1994), this mutation adapts and checks each mutation event on whether it increases the solution fitness or not. If so, it accepts the mutation, else it continues to find another beneficial allele to mutate towards. In many ways this is similar to a local "gradient" search heuristic which searches the local solution space neighbourhoods for improvements.

This local search can be thought of as a kind of learning that happens during the lifetime of the solution (i.e. within a single generation). The learnt "behaviour" can also improve an individual's chance of survival as well as their genetic makeup. The consideration is whether such learnt behaviour should be encoded in the genotype or not, or only used to increase the fitness of that solution temporarily for that generation. In our case, we chose to encode the learnt improvements through a mutation in the genotype as these beneficial mutations could then be passed on to the children in the next generation.

The mutation goes through the RS list, and if exhausted, uniformly random chooses a 0-frequency allele to mutate towards if it increases the fitness of the solution.

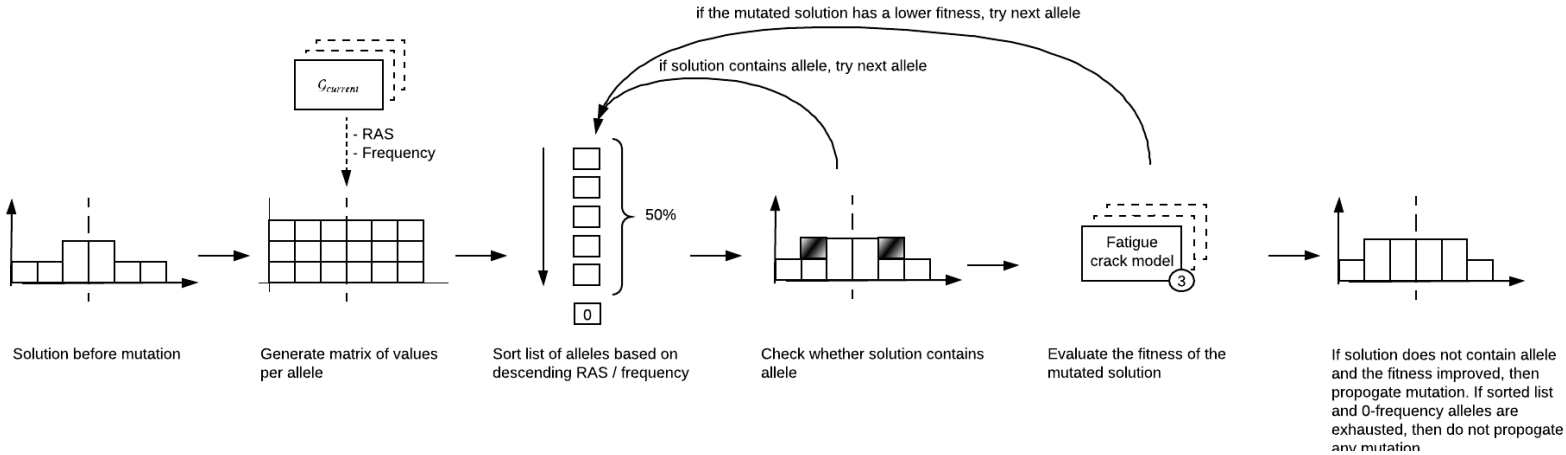


Figure 17 Schematic overview of relative strength allele adaptation (RSAA-M) mutation

Mutation 3. | Relative strength memory (RSM-M)

Concepts included are PRAS signalling and solution memory

A role of mutation is considered to be that of maintaining genetic diversity in the population. RSMM adds a population level diversity mechanism to the RSM heuristic, as every mutated solution is checked for duplicates against the current population. If a duplicate is found, the mutation is rejected and another mutation is performed, up until a maximum of 25 attempts, after which mutation does not take place. RSM mutation uses the same mutation heuristic as RS-M, but adds another layer of diversity preservation

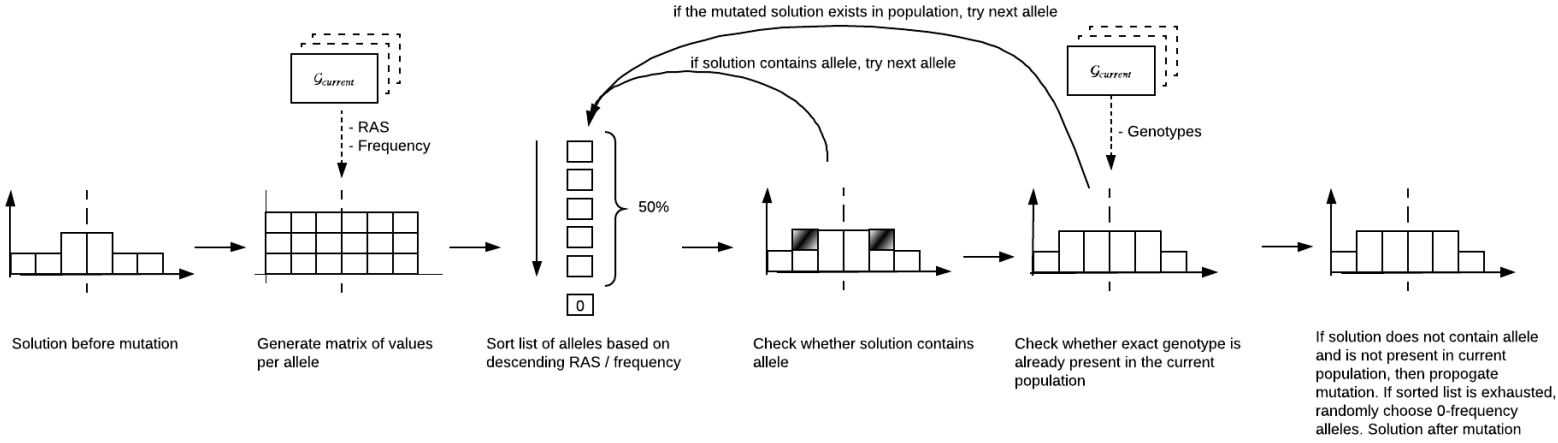


Figure 18 Schematic overview of relative strength memory (RSM-M) mutation

Mutation 4. | Relative strength adaptive allele strong solutions (RSAASS-M)

Concepts included are PRAS signalling, mutation filtering and selection bias

RSAASS-M introduces an a-symmetry in mutation rate for solutions based on their relative fitness rank in the population. If a solution is part of the top 50% of solutions in the current population, the mutation rate for this solution increases to 75%.

This mutation is largely equivalent to RSAA-M, only that the mutation probability for the upper 50% of the population in terms of fitness is significantly higher. This implies that the positive mutations are propagated more specifically on the strongest solutions in the population and therefore the survival probability of this allele after mutation is expected to be higher.

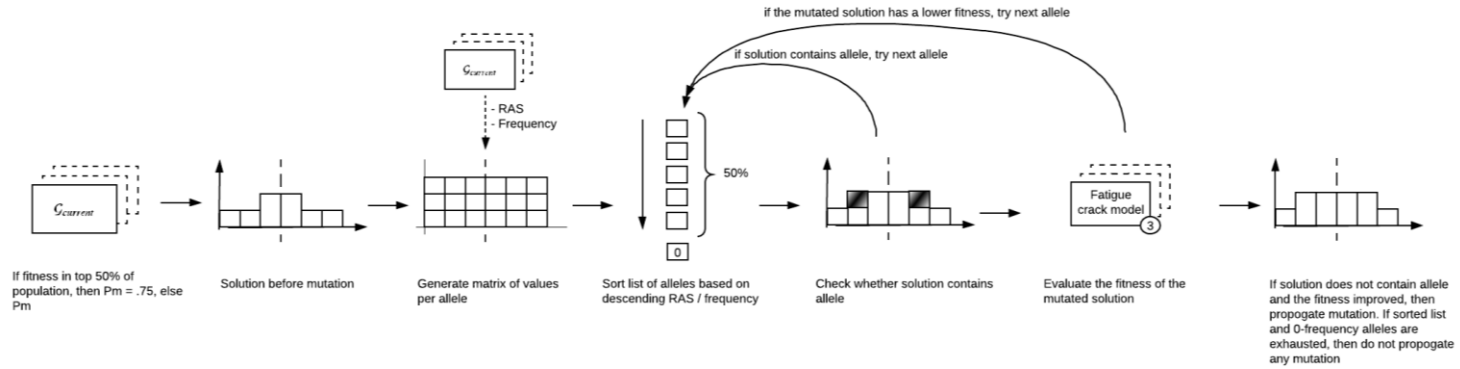


Figure 19 Schematic overview of relative strength adaptive allele strong solutions (RSAASS-M) mutation

Mutation 5. | Relative strength adaptive allele strong solutions enumeration (RSAASSEUE-M)

Concepts included are PRAS signalling, mutation filtering and selection bias

The difference in this algorithm is that it includes not only 0-frequency alleles, but also the ubiquitous alleles. Furthermore, it enumerates each of the ubiquitous and 0-frequency alleles, in order to prevent sampling to concentrate on a number of alleles which do not yield improved fitness.

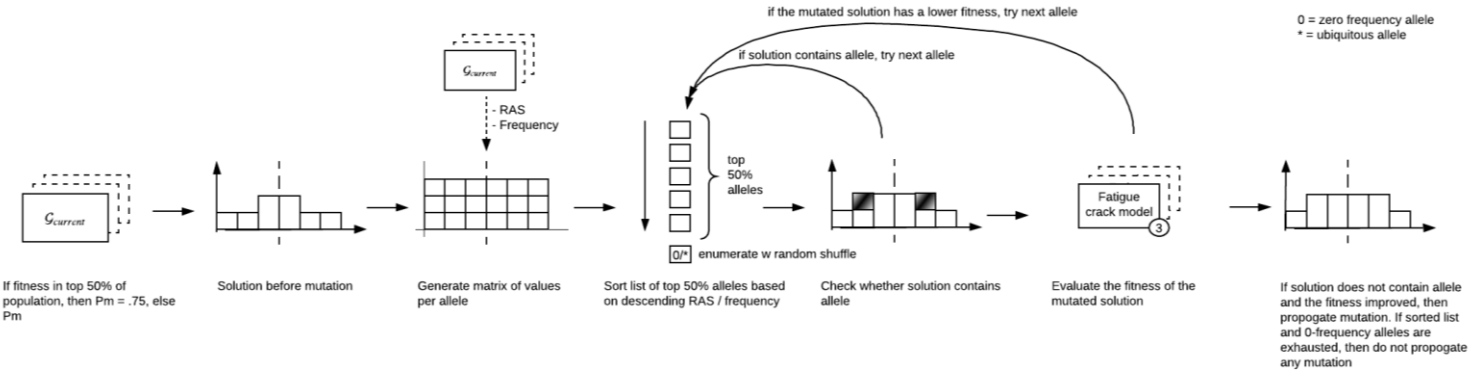


Figure 20 Schematic overview of relative strength adaptive allele strong solutions enumeration (RSAASSEUE-M) mutation

If sampled allele is present, then remove it, if not present, then add it. As such the GA does not have any explicit knowledge of the allele end-state.

Important to note is that with crossover, the population can naturally grow as each crossover event creates 2 offspring, while mutation changes a single solution and thus creates a single offspring. However, when selection splits the population in half in terms of fitness and removes the bottom half, it must be resized to the original population size. Whenever only mutation in this research, the surviving population is duplicated in order to return to the full population size.

4.5.3. CUSTOM INITIALISATION METHODS AND POPULATIONS

As described in the GA introduction, first, a set $\mathcal{G}_{init}$ of initial solutions s_i where $i, \dots, N_{pop}$ needs to be generated. The experiments in this research have initial populations that can either be generated at random and predetermined. If not random, the populations can either be uniform or diverse. The below description details how the initial populations are created in each case.

Random diversified (RD-I)

For the majority of the experiments the populations are generated at random and ensured to be as diverse as possible. It was ensured that the avg initial PRAS did not deviate too much from 50% in order not to bias the results based on better or worse initial conditions. The initial avg PRAS ended up being 58%. Further experimentation on the effect of strong and weak initial populations would provide more insight into the effect this initial condition. The initialisation process in Figure 21 repeats until the population size has been reached.

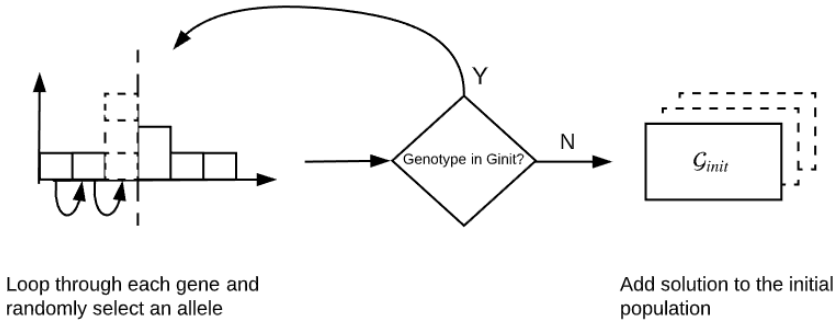


Figure 21 Schematic view of the random diversified initialisation heuristic

Determined diverse (DD-I) and uniform (DU-I)

Initial populations of 10 solutions were created by hand picking solutions for experiments with γ = 15 which required a uniform/diverse and a low/high initial PRAS. These solutions were designed for the 3rd objective function ($F_3(s) = \frac{N}{A^m}$), where the optimal solution would become known through brute force enumeration of all possible solutions. Please note that the GA itself would not have received any direct information on the optimal solutions and that this brute force method is for evaluation purposes only. These optimal solutions are presented and elaborated upon in the first section of the results chapter.

In the definition of non-random, pre-determined populations there was no particular consideration about which solution to give which allele values except for that the population matrix in general should result in a low or high initial PRAS level. In the diverse populations consideration was made to make the populations diverse enough while biasing them towards a high or low initial PRAS.

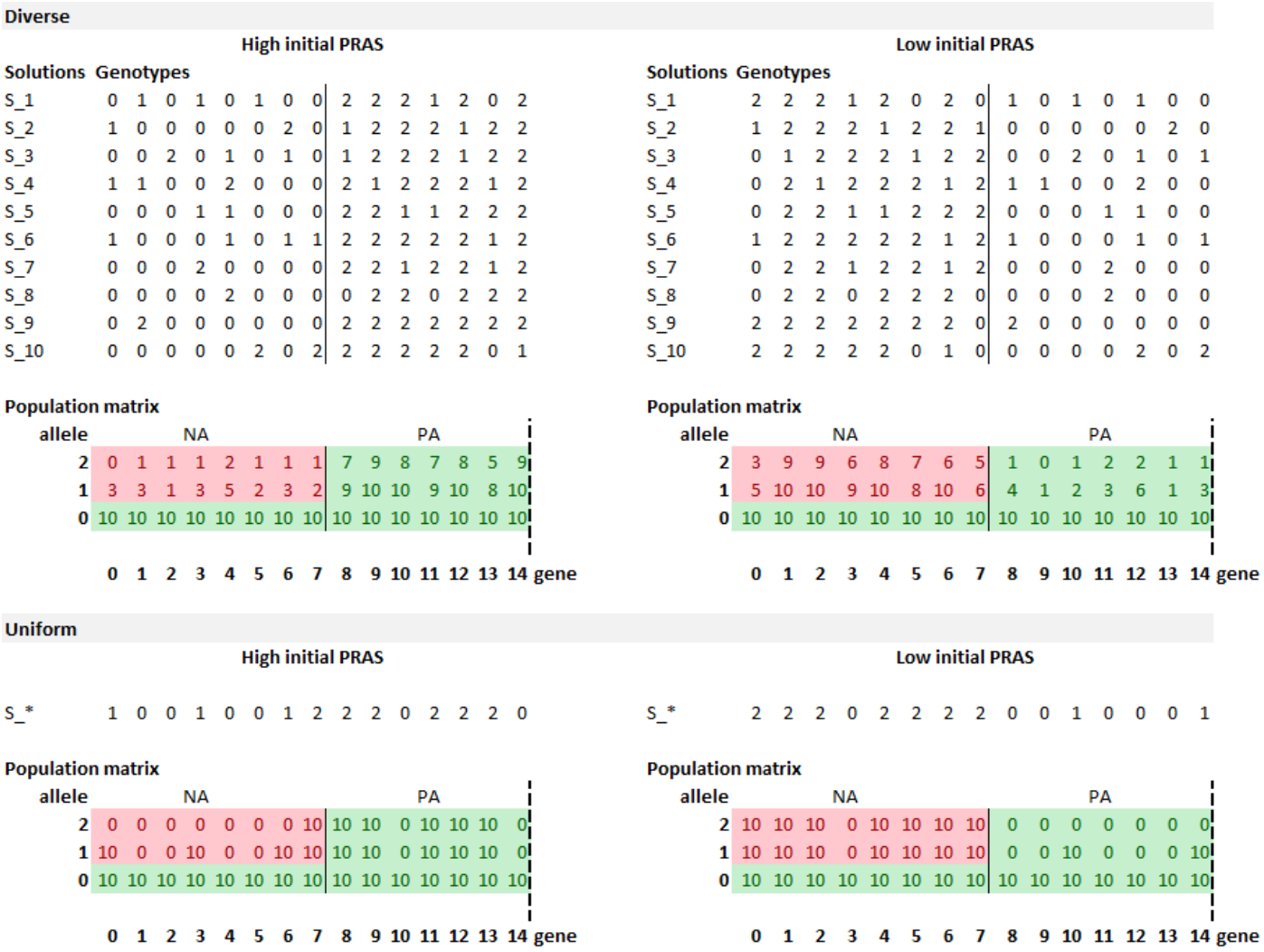


Figure 22 Initial populations genotype matrix consisting of diverse and uniform populations, and high or low initial PRAS strength

A careful look at the population matrix of both uniform low and high initial PRAS will show the shapes of the initial population plates. Since they are uniform, each solution has the same shape and thus the aggregate population matrix will be a singular shape as well.

4.6. GA OPTIMISATION APPROACHES

4.6.1. EXPERIMENTAL MATRICES

Considering that the overall GA optimisation framework with its design problem, analytical model and (custom) heuristics have been defined, a systemic approach to experimenting with this framework needs to be detailed. The experimental matrices in Table 9, Table 10 and Table 11 provide an overview of the experiments in this research. Table 9 concerns experiments for heuristics with varying objective function and granularity where combinations with 'X' did not end up being reported in the results & analysis due to the lack of informative value as the optimal solution ended up being trivial – a full block of material.

Table 9 Experimental matrix for testing GA designs for varying objective function and granularity

Problem parameters			**Crossover only**			**Mutation only**						**Crossover and mutation**					
$F(s)$	γ	G	SPC	TPC	UC	UM	RSM	RSMM	RSAAM	RSAASSM	RSAASSEUEM	UC UM	UC RSM	UC RSMM	UC RSAAM	UC RSAASSM	UC RSAASSEUE M
N	5	20	X	X	X	-	-	-	-	-	-	-	-	-	-	-	-
	15		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
$\frac{N}{A^m}$	5		-	-	-	X	X	X	X	-	-	-	-	-	-	-	-
	15		EX-1	EX-2	EX-3	EX-11	EX-12	EX-13	EX-14	EX-15	EX-16	EX-17	EX-19	EX-18	EX-20	EX-21	EX-22
	150		-	-	-	-	-	-	-	-	-	EX-64	EX-63	-	EX-61	EX-62	EX-65
		40	-	-	-	-	-	-	-	-	-	-	-	-	-	EX-66	EX-67

Table 10Table 9 concerns experiments for heuristics initial populations with a variation on mutation rate. Its primary function what to provide understanding on how the initial population composition affects the optimisation end results under various GA design conditions.

Table 9 Table 11 concerns experiments for various GA designs with crossover and mutation and mutation rates to understand the influence on the optimisation performance.

Table 10 Experimental matrix for testing GA design for varying initial populations and mutation rate

Parameters & heuristics			**Crossover**			**Crossover and mutation**					
Initialisation	Initial PRAS	Pm	SPC	TPC	UC	UC UM	UC RSM	UC RSMM	UC RSAAM	UC RSAASSM	UC RSAASSEUE M
DD-I	High	0.1	EX-8	EX-7	EX-6	EX-37	EX-28	-	EX-31	EX-33	EX-34
DD-I	Low		EX-9	EX-5	EX-4	EX-36	EX-29	-	EX-30	EX-32	EX-35
DU-I	High		-	-	-	-	-	-	-	-	-
DU-I	Low		-	-	-	EX-24	EX-25	-	EX-26	EX-27	EX-23
		0.2	-	-	-	EX-40	EX-39	-	-	-	EX-38

Table 11 Experimental matrix for testing GA designs for varying mutation rate

Parameter **Crossover and mutation**

Mutation rate	UC UM	UC RSM	UC RSMM	UC RSAAM	UC RSAASSM	UC RSAASSEUE M
0.05	EX-44	EX-41	-	EX-42	EX-43	EX-45
0.1	EX-49	EX-46	-	EX-47	EX-48	EX-50
0.15	EX-54	EX-51	-	EX-52	EX-53	EX-55
0.2	EX-59	EX-56	-	EX-57	EX-58	EX-60

4.6.2. DEFAULT HYPERPARAMETERS

In the GA a number of hyperparameters must be defined (Back, et al., 2000). The default settings for various parameters, unless stated otherwise, are shown in Table 12. A population size of 20 was chosen in relation to the number of genes and alleles and iteratively determined by running experiments. A relative small population would unnecessarily limit the optimisation, while a large population would not provide a realistic constraint often present in real optimisations. Larger population generally require more computational effort and one does not want have unnecessarily large populations if the same results can be achieved with a smaller population. The number of generations was chosen following a similar approach.

The number of trials is important as a stochastic process is investigated. Thus, the results of the optimisation are different after each trial (or run). The number of trials was iteratively chosen by determining whether the average results with an additional trial would continue to change much. The number of trials was minimized since each additional trial requires significant computational time. Tournament size was chosen based on what was often found in literature (Back, 1994)

Table 12 Default genetic algorithm hyperparameter settings unless stated otherwise

Variable	**Npop**	**Pm**	**Pc**	**Rs**	**T**	**Generations**	**Trials**
Value	20	0.1	1	0.5	0.3	20	10

5. RESULTS & ANALYSIS

The results of this research can be divided into several components. First, the fatigue crack growth model is verified with experimental data. Second, different type of objective functions for the optimisation of thickness distribution are investigated mathematically. Third, the GA variations are tested on ranges of granularity γ. Fourth, a sensitivity study on the GA parameters is performed.

5.1. HEURISTIC CONCEPTS

5.1.1. CROSSOVER ONLY

Experiment trials with UC had higher avg. end-state PRAS values than SPC and TPC (Figure 23). UC trials reached an average of 0.93, while SPC and TPC improved to 0.82 and 0.86 respectively. The avg PRAS effective for each experiment followed a nonlinear path over the generations with most of the improvement taking part in the first 10 generations.

Experiment trials with UC had the highest GA objective function score of 23.0, while TPC and SPC concluded with 9.7 and 6.0 respectively. With respect to GA function rates, none of the experiments was able to introduce nor eliminate any alleles in(to) the population as demonstrated by the introduction and elimination rate with a value of 0.

Progression rates (0.17-0.18) were balanced among the experiments, while larger differences were observed for the suppression rates (0.13-0.16). UC demonstrated the lowest suppression rate of 0.13 PRAS per suppression event. Most progression events were observed in the UC experiment trials, as well as for the number of ubiquity events. The inverse is demonstrated for the suppression and extinction events.

The cumulative number of progression events is larger than the suppression events for each GA. A similar observation can be made for the ubiquity events as compared to the extinction events. Introduction and elimination events are zero for all GAs.

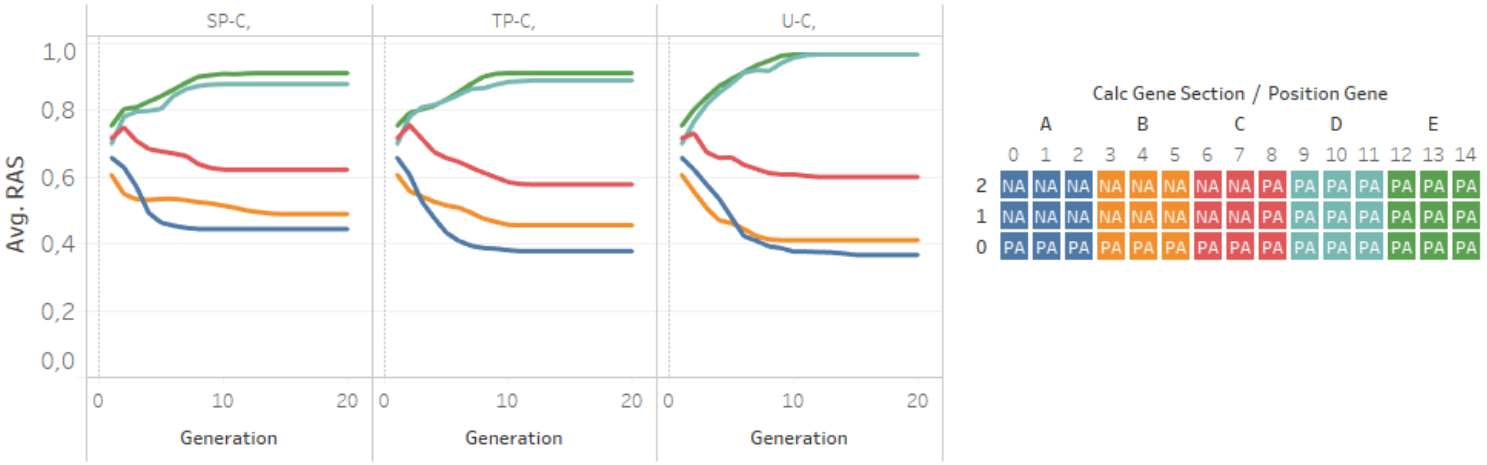
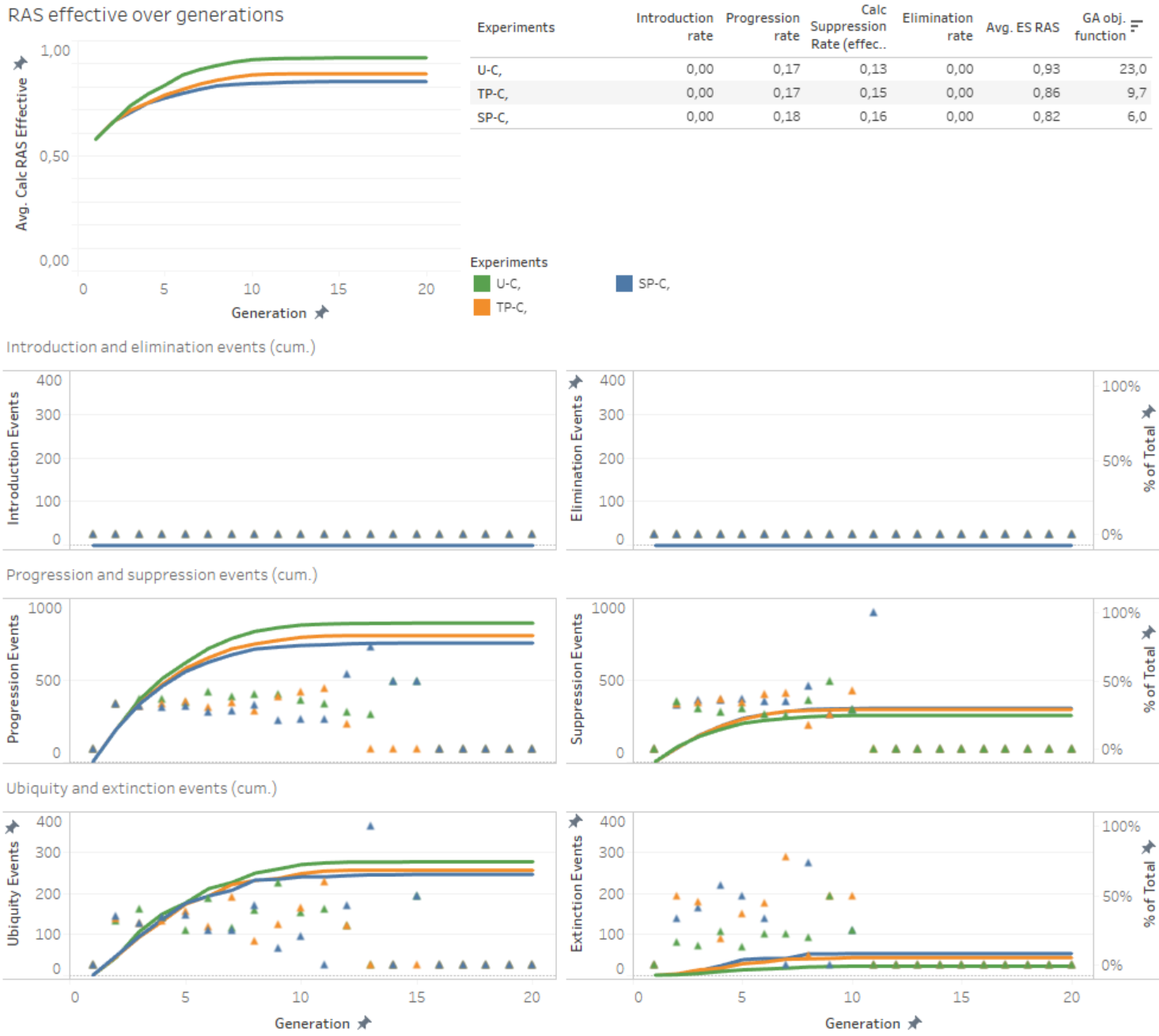


Figure 23 Avg. PRAS over generations for each crossover only experiment

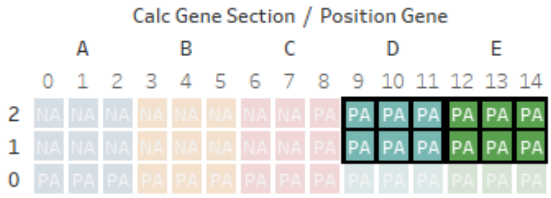


Experiments	Introduction rate	Progression rate	Calc Suppression Rate (effec..	Elimination rate	Avg. ES RAS	GA obj. function
U-C,	0,00	0,17	0,13	0,00	0,93	23,0
TP-C,	0,00	0,17	0,15	0,00	0,86	9,7
SP-C,	0,00	0,18	0,16	0,00	0,82	6,0

Figure 24 Avg. PRAS effective over generations (top left), GA function rates (top right) and GA function events over generations (bottom) per crossover only GA

Closer inspection on the allele and gene section level of detail (Figure 23) demonstrated that UC trials were able to progress the gene sections A, B, D and E more in terms of PRAS than SPC and TPC. The PA gene sections (D,E) reached levels of avg PRAS of 0.98, while the NA gene section (A, B) decreased far less in PRAS to levels as high as 0.41.

Isolating the gene sections D and E (Figure 25) revealed that these alleles produced more progression and ubiquity events, as well as less suppression and extinction events. What stands out is the large relative difference in extinction events related to UC trails in comparison to SPC and TPC trials.



Experiments	Progression events	Ubiquity events	Suppression events	Extinction events	Extinction Δ from baseline
U-C	352	114	120	6	-68%
TP-C	303	102	136	18	-5%
SP-C (baseline)	315	101	137	19	0%
Grand Total	970	317	393	43	

Figure 25 Overview of GA function events per experiment with a change in extinction events w.r.t. the baseline

5.1.2. MUTATION ONLY

Experiment trials with mutation only GAs demonstrated a larger spread in avg. end-state PRAS as compared to crossover only GAs (Figure 27). Trials with RSAASSEUE-M consistently resulted in the highest achievable avg. PRAS effective of 1.0. Other mutation heuristics RSAASS, RSAA and RS showed little spread in their avg end-state PRAS (0.81 – 0.89), providing improvements between 42% and 56% of the initial avg. RAS. RSM demonstrated the lowest avg. end-state PRAS of 0.71, an improvement of only 25%. RSAASSEUE-M reached the highest GA objective function score of 55.9 while the RSAASS-M ended second place with a score of 10.8.

The trials with UM contain an inexplicable error in the initial population. Even though the initial population is the same, the initial PRAS in the experimental data is not equivalent to that of the other initial populations. Nonetheless,

Each of the mutation heuristics was able to perform all of the GA functions, except for RSAASSEUE-M which demonstrated an elimination rate of 0 as it did not produce any elimination events. With respect to introduction, RSAASSEUEM demonstrated the highest rate of 0.35. The spread amongst introduction and elimination rates was relatively large (0.11 – 0.35) as compared to the other GA functions, where the spreads were 0.17 – 0.23 and 0.16 – 0.20 for progression and suppression respectively.

Each of the experiments demonstrated a linear trend for the cumulative number of introduction events between 0 < g < 20. UM demonstrated the same trend for each of the other functions, however most other experiments seem to reach plateaus generally visible at g = 5. The exception to this is RSM which showed linear trends for other functions more similar to UM.

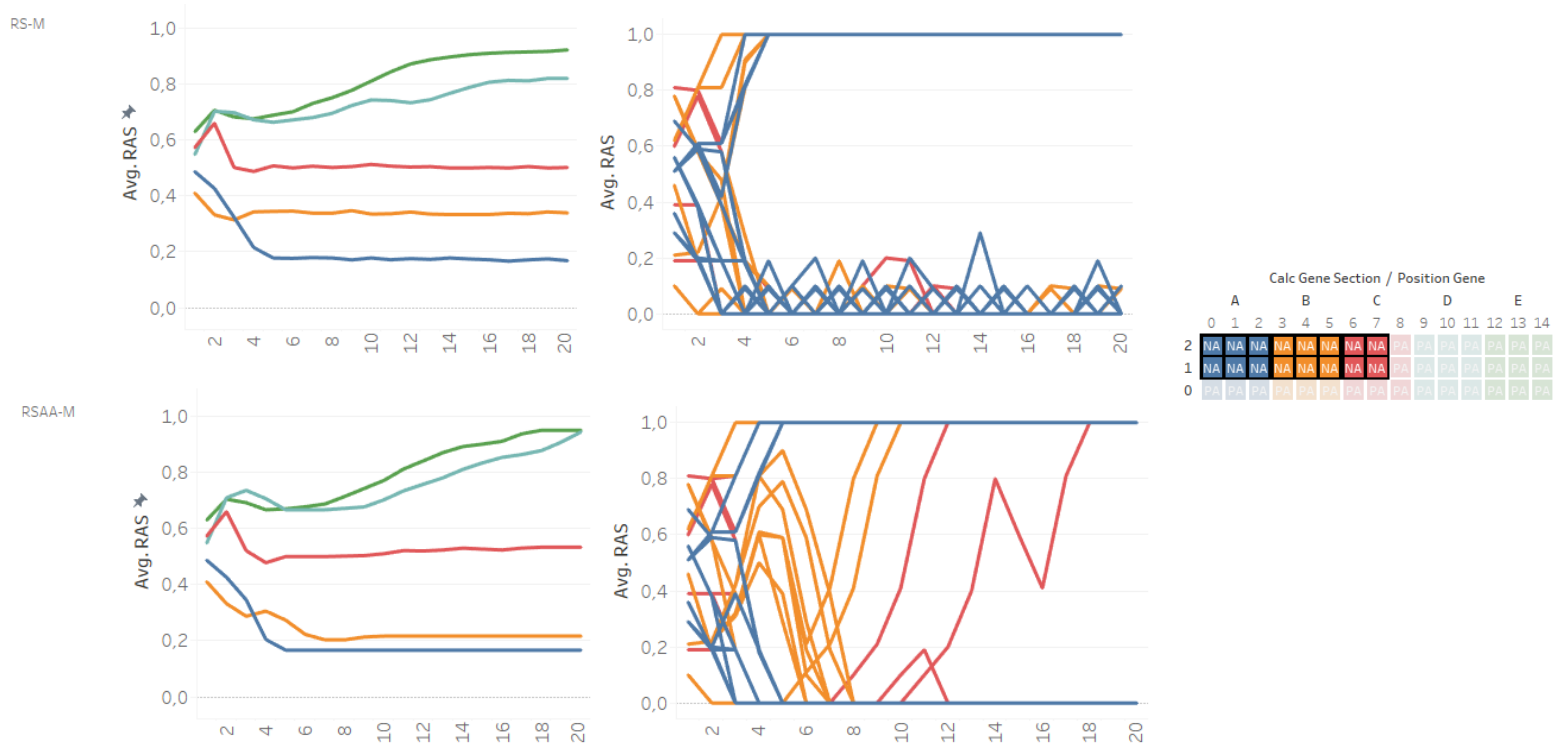
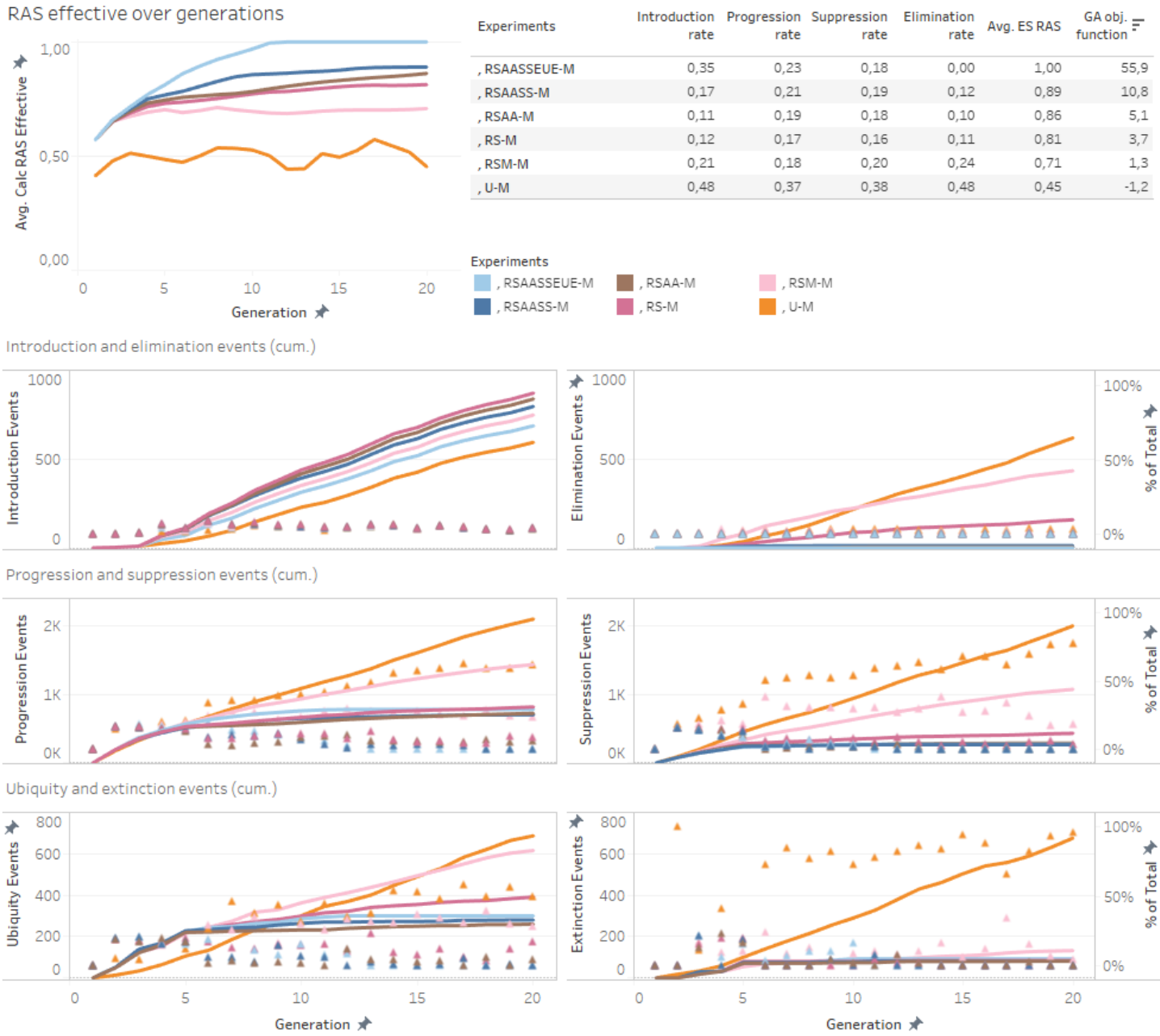


Figure 26 Avg PRAS over generations per gene section overall (left) and isolated for NAs (right) for RS and RSAA

Further investigation into the gene sections and single runs per allele a number of patterns were observed. First, RSM demonstrated that it could increase the avg RAS per gene section within 0 < g < 5 for the gene



Experiments	Introduction rate	Progression rate	Suppression rate	Elimination rate	Avg. ES RAS	GA obj. function
, RSAASSEUE-M	0,35	0,23	0,18	0,00	1,00	55,9
, RSAASS-M	0,17	0,21	0,19	0,12	0,89	10,8
, RSAA-M	0,11	0,19	0,18	0,10	0,86	5,1
, RS-M	0,12	0,17	0,16	0,11	0,81	3,7
, RSM-M	0,21	0,18	0,20	0,24	0,71	1,3
, U-M	0,48	0,37	0,38	0,48	0,45	-1,2

Figure 27 Avg. PRAS effective over generations (top left), GA function rates (top right) and GA function events over generations (bottom) per mutation only GA

section A (NA) and D (PA) seen by the respective decrease and increase in average RAS within that range (). A similar but much less pronounced optimisation effect can be seen for the other gene sections.

The main limitation of RS is that it continued to introduce NA alleles into the population which can be seen by the relative high number of introduction events on NA gene sections A, B and C (Figure 29). A secondary limitation is that RS was not able to eliminate any NA alleles of gene sections A, B and C that had become ubiquitous in the population.

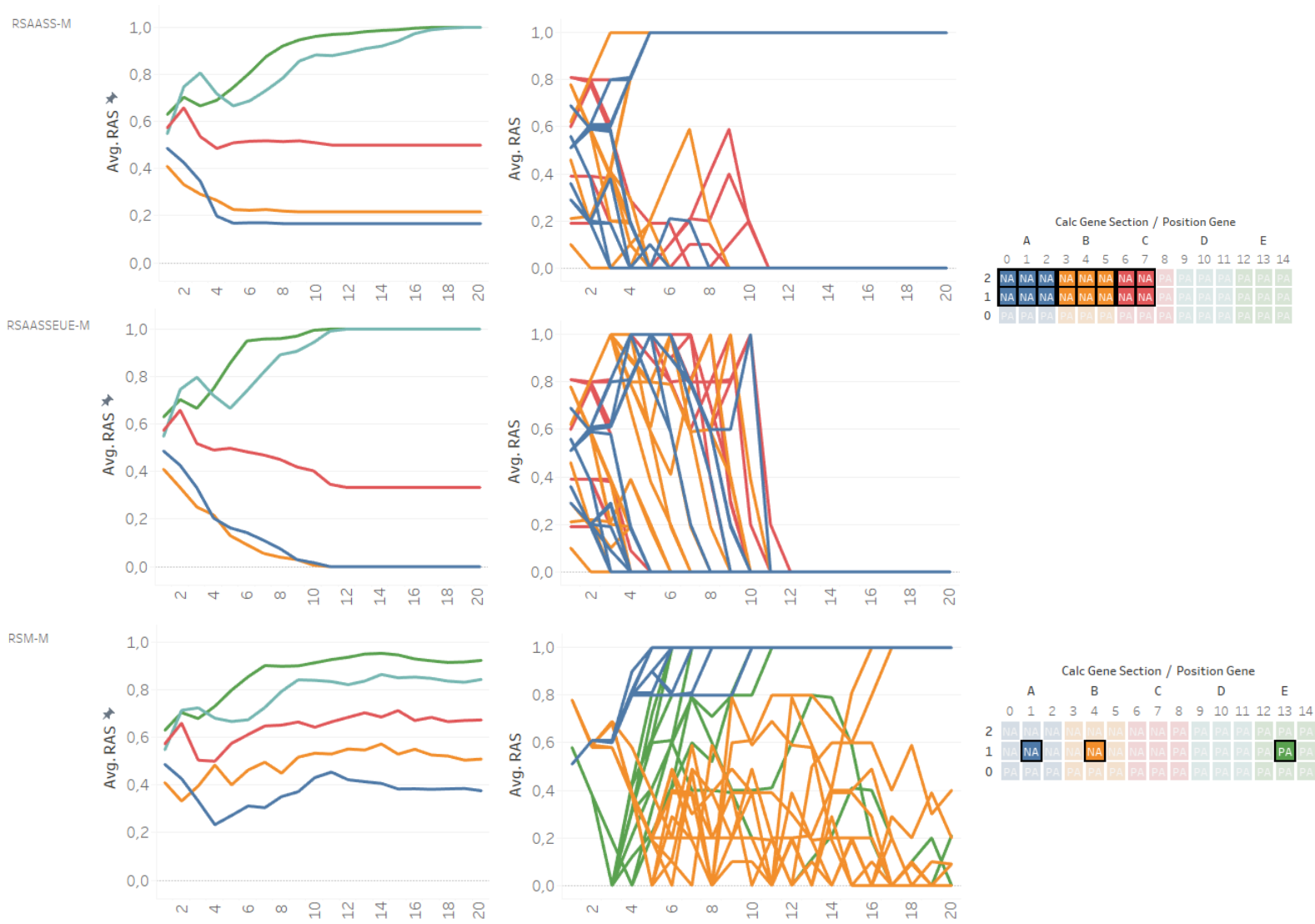


Figure 28 Avg PRAS over generations per gene section overall (left) and isolated for NAs (right) for RSAASS, RSAASSEUE and RSM

RSAA demonstrated improved end-state RAS for gene sections B and D. The events produced by these trials indicate improvements on the main limitation of RSM, as the number of introduction events on NA alleles nearly disappeared. On the other hand, RSAA trials did not achieve more elimination events of ubiquitous NAs on gene sections A, B and C (Table X).

RS-M

	A	B	C	D	E	Grand Total
Introduction	66	54	28	15	18	181
Progression	105	114	95	173	146	633
Ubiquity	10	20	30	51	58	169
Elimination	0	0	0	4	6	10
Suppression	207	143	120	92	54	616
Extinction	115	90	57	24	20	306

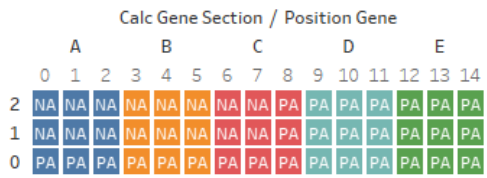


Figure 29 Total number of events per GA function, gene section and allele end-state for RS-M

Furthermore, the extinction events for NA gene sections A, B and C reduced as these NA alleles were not reintroduced and eliminated again, which would otherwise have counted towards extinction events

RSAA-M

	A	B	C	D	E	Grand Total
Introduction	0	1	4	23	19	47
Progression	41	61	85	199	148	534
Ubiquity	10	13	32	53	57	165
Elimination	0	0	0	0	0	0
Suppression	140	113	92	81	52	478
Extinction	50	48	32	24	22	176

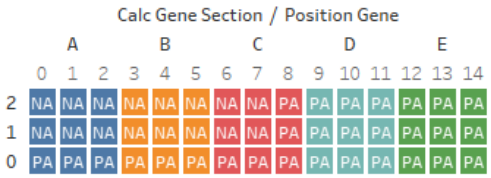


Figure 30 Total number of events per GA function, gene section and allele end-state for RSAA-M

Investigating the PA gene sections D and E for RSAA shows that there were no more elimination events, while event related to the other GA functions remained close to the values produced by RS (Figure below).

Another observation is that while NAs demonstrated less introduction events, when they did occur, those alleles more often progressed and became ubiquitous as compared to the RS trials (Figure 31). Investigating the trials on those alleles shows that these trials correlate with the average PRAS of various PAs.

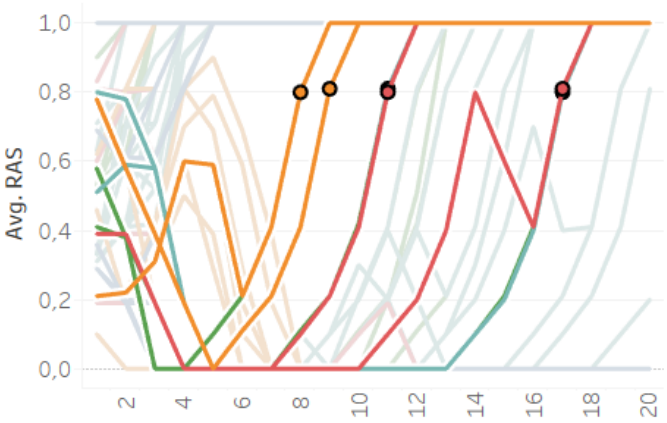


Figure 31 Hitchhiking of negative alleles (orange, red) with positive alleles (teal, green) in RSAA-M

RSAASS trials showed a slight increase in introduction events on NA gene sections A, B and C (Figure 32), yet these introductions did not correlate with an increase in ubiquitous NAs. Furthermore a difference with RSAASS trials was the increased introduction and progression rates (Figure 27) compared to RSAA and RS. Thus, for every introduction or progression event, more PRAS was changed. A notable observation is that RSAASS was able to fully optimise PA gene sections D and E to avg RAS = 1 within $0 < G < 20$ (Figure 28).

RSAASS-M

	A	B	C	D	E	Grand Total
Introduction	2	4	7	26	22	61
Progression	47	54	83	198	141	523
Ubiquity	10	13	30	60	60	173
Elimination	0	0	0	0	0	0
Suppression	128	108	94	74	42	446
Extinction	52	51	37	26	22	188

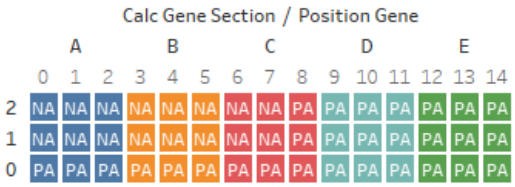


Figure 32 Total number of events per GA function, gene section and allele end-state for RSAASS-M

Figure 28 demonstrates the inability to eliminate ubiquitous NAs remains present in RSAASS. Trails with RSAASSEUE show that the GA is able to consistently convergence on the optimal solutions with an avg. PRAS = 1. The gene level shows how the ubiquitous NAs are eliminated and suppressed to RAS = 0 until no ubiquitous NAs were left (Figure 28).

RSAASSEUE-M

	A	B	C	D	E	Grand Total
Introduction	0	0	0	27	23	50
Progression	47	45	83	165	124	464
Ubiquity	11	14	38	60	60	183
Elimination	11	14	18	0	0	43
Suppression	180	159	122	81	48	590
Extinction	60	60	40	27	23	210

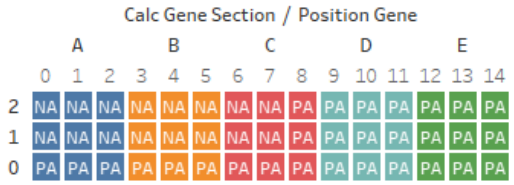


Figure 33 Total number of events per GA function, gene section and allele end-state for RSAASSEUE-M

RSM trials were not able to effectively optimise the design problem with an average end-state PRAS of 0.71 (Figure 28). While RSM hardly eliminated any alleles, it demonstrated a lot of variation in PRAS within the bounds of 0 < PRAS < 0.9 (). The same observation can be made in the number relatively high number of introduction, progression, suppression and extinction events (Table below).

RSM-M

	A	B	C	D	E	Grand Total
Introduction	187	152	94	34	26	493
Progression	314	303	259	217	167	1.260
Ubiquity	18	28	38	48	55	187
Elimination	0	1	0	0	1	2
Suppression	422	350	246	149	82	1.249
Extinction	216	171	106	39	29	561

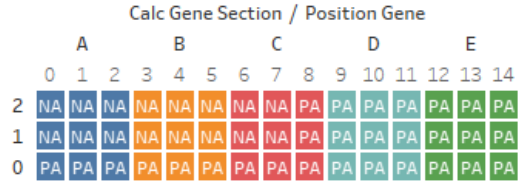


Figure 35 Total number of events per GA function, gene section and allele end-state for RSM-M

U-M

	A	B	C	D	E	Grand Total
Introduction	128	128	122	118	110	606
Progression	397	402	412	435	401	2.047
Ubiquity	123	144	136	142	138	683
Elimination	106	130	121	122	128	607
Suppression	438	412	407	394	401	2.052
Extinction	142	140	139	135	124	680

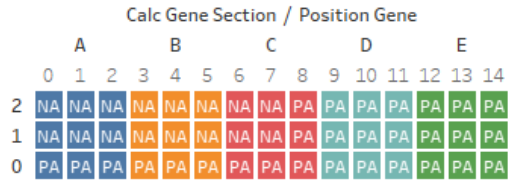
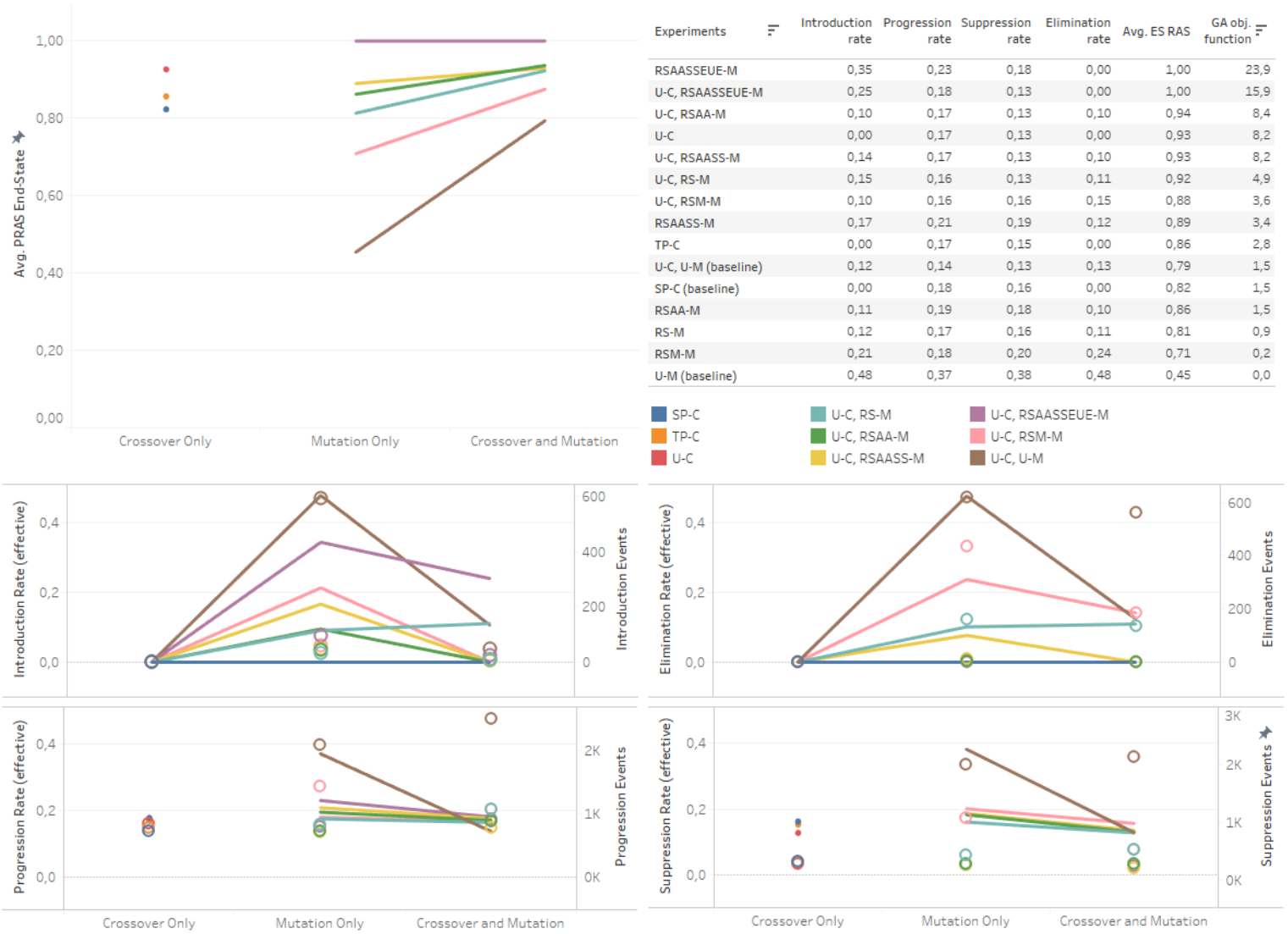


Figure 34 Total number of events per GA function, gene section and allele end-state for U-M

5.1.3. CROSSOVER AND MUTATION

In comparison with the mutation only and crossover only experiments, the trials with both crossover and mutation heuristics made it possible for more versions of the GA to reach higher levels of end-state average PRAS (Figure 36). Each mutation only GA except for RSAASSEUE increased their end-state RAS through the inclusion of UC. However, UC alone reached higher end-state PRAS than many of the crossover and mutation GAs.

Furthermore, SPC and TPC outperformed mutation only GAs RS, RSM and RSAA in the levels of end-state RAS. The GA objective function score of RSAASSEUE decreased with the addition of UC from 23.9 to 15.9, a decrease of 33% (Figure 40). The standard GAs in any case were not able to reach levels higher than 0.79 avg end-state RAS (UC UM).



Experiments	Introduction rate	Progression rate	Suppression rate	Elimination rate	Avg. ES RAS	GA obj. function
RSAASSEUE-M	0,35	0,23	0,18	0,00	1,00	23,9
U-C, RSAASSEUE-M	0,25	0,18	0,13	0,00	1,00	15,9
U-C, RSAA-M	0,10	0,17	0,13	0,10	0,94	8,4
U-C	0,00	0,17	0,13	0,00	0,93	8,2
U-C, RSAASS-M	0,14	0,17	0,13	0,10	0,93	8,2
U-C, RS-M	0,15	0,16	0,13	0,11	0,92	4,9
U-C, RSM-M	0,10	0,16	0,16	0,15	0,88	3,6
RSAASS-M	0,17	0,21	0,19	0,12	0,89	3,4
TP-C	0,00	0,17	0,15	0,00	0,86	2,8
U-C, U-M (baseline)	0,12	0,14	0,13	0,13	0,79	1,5
SP-C (baseline)	0,00	0,18	0,16	0,00	0,82	1,5
RSAA-M	0,11	0,19	0,18	0,10	0,86	1,5
RS-M	0,12	0,17	0,16	0,11	0,81	0,9
RSM-M	0,21	0,18	0,20	0,24	0,71	0,2
U-M (baseline)	0,48	0,37	0,38	0,48	0,45	0,0

Figure 36 Avg. end-state PRAS (top left), GA function rates and scores (top right) and GA function rates and events (bottom) per GA grouped by crossover, mutation or both

Generally, the addition of UC seemed to reduce the number of introduction and elimination events, as well as their rates (Figure 36). Exceptions to this are UC UM and UC RS, where this effect is much smaller, or even

negligible. An important observation was that in every case the addition of UC decreased the number of generations required for the experiments to reach their avg PRAS plateaus (Figure 36). This observation correlates with the fact that the addition of UC led to higher end-state levels of PRAS.

UC, RSAASSEUE experiment trails were the only combination of crossover and mutation to achieve an average end-state RAS of 1.0 consistently with a GA objective function score of 15.9. According to our GA objective function, it did so less efficiently as RSAASSEUE which had a score of 23.9.

U-C, RSAASS-M

	A	B	C	D	E	Grand Total
Introduction	1	0	0	1	2	4
Progression	43	50	88	170	149	500
Ubiquity	3	7	31	60	60	161
Elimination	0	0	0	0	0	0
Suppression	190	142	100	34	28	494
Extinction	58	53	29	1	2	143

U-C, RSAASSEUE-M

	A	B	C	D	E	Grand Total
Introduction	0	0	1	3	4	8
Progression	46	53	99	174	149	521
Ubiquity	1	6	30	60	60	157
Elimination	1	6	10	0	0	17
Suppression	202	185	169	46	42	644
Extinction	60	60	41	3	4	168

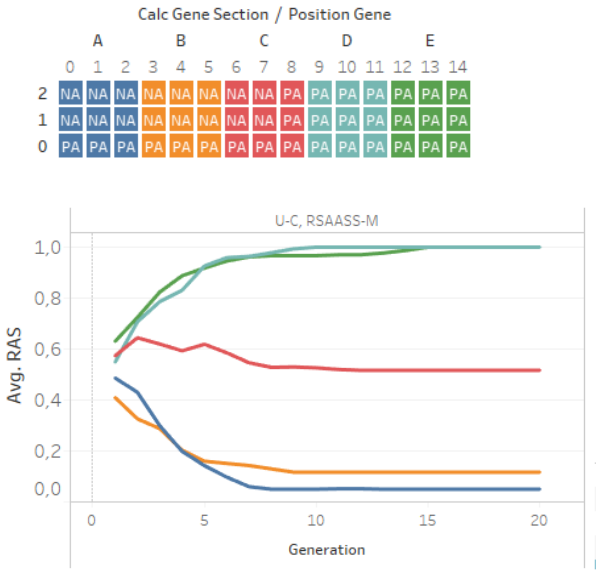


Figure 37 Total number of events per GA function, gene section and allele end-state for UC with RSAASS-M or RSAASSEUE-M

UC,RSAA and UC,RSAASS came close to each other with a GA objective function score of 8.4 and 8.2 respectively. The introduction rate of UC,RSAASS remained higher, however this increase did not propagate to the progression and suppression rates as it did with the mutation only GA. The gene section level shows that the inability to eliminate ubiquitous NAs was the limitation for UC RSAASS to further progress on avg PRAS. A comparison between events in Figure 37 shows this difference in elimination events in gene section A, B and C.

U-C, U-M

	NA				PA				Grand Total
	A	B	C	Total	C	D	E	Total	
Introduction	143	123	45	311	0	14	5	19	330
Progression	380	413	292	1.085	144	514	491	1.149	2.234
Ubiquity	5	4	23	32	36	111	138	285	317
Elimination	4	4	20	28	36	94	123	253	281
Suppression	494	511	345	1.350	166	437	441	1.044	2.394
Extinction	158	143	56	357	1	15	5	21	378

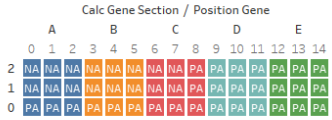


Figure 38 Total number of events per GA function, gene section and allele end-state for UC,UM

The addition of UC to UM rates led to more frequent events of smaller size in terms of PRAS. The data in Figure 38 shows how the SGA (U-C, U-M) reintroduces and progresses NAs, while repeatedly eliminating PAs. Results of this experiment further showed that in the progression events on PA gene sections was 6% (1149 vs 1085) higher and suppression events on NA gene sections were 30% higher than on PA sections. As such, this means that PA genes progressed more often in PRAS and NA were suppressed more often in PRAS.

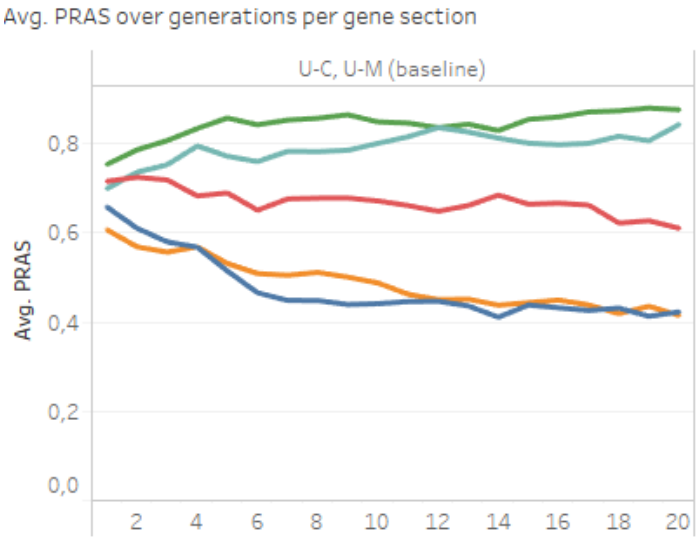


Figure 39 Avg PRAS per gene section over generations for UC, UM

Figure 39 shows how these events correlate with a decreasing avg PRAS for NA gene sections A and B, while the avg PRAS for PA gene sections D and E are increasing. However, the end-state PRAS of these gene section are far from the optimal states. Especially the NA gene sections seem further away from the optimal (0.4 PRAS), even though the change in PRAS from generation 0 to 20 is the largest.

The difference in avg PRAS effective per GA experiment group with either mutation only or both crossover and mutation is shown in Figure 40. UC primarily results in an increase of end-state PRAS except for UC, RSAASSEUEM, which achieve the same result without UC.

Furthermore, UC seems to increase the PRAS increase in earlier stages of the optimisation, roughly within the generations range of $0 < G < 10$. This effect is still present with RSAASSEUEM, yet it is the smallest effect among all GAs.

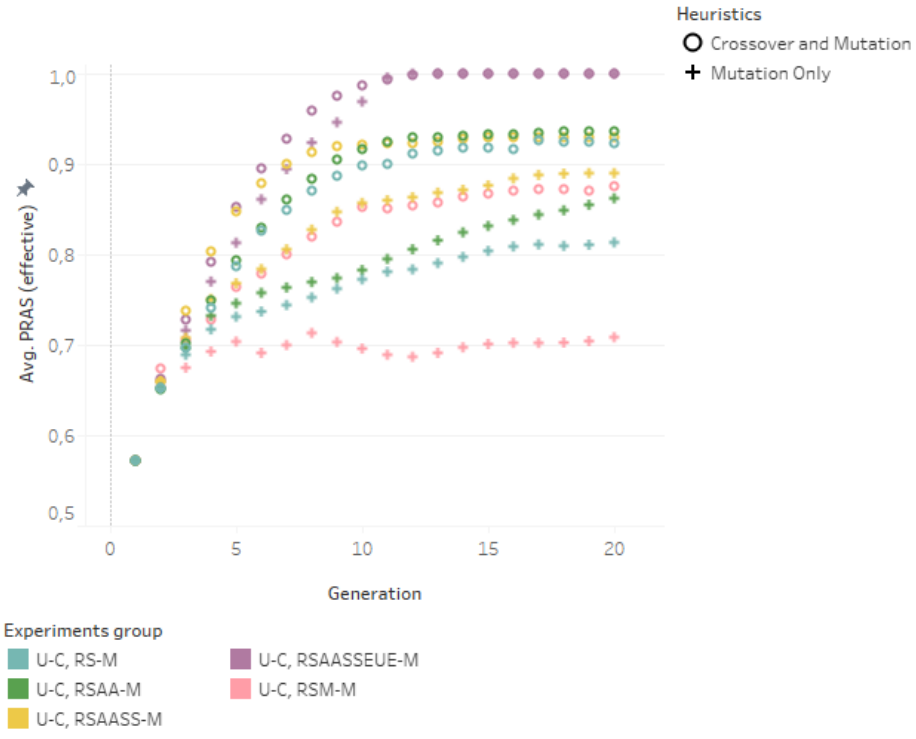


Figure 40 Optimisation path comparison between mutation only and crossover with mutation. Axes shortened to better indicate the differences for readability purposes

5.2. INITIAL POPULATION

5.2.1. HIGH VERSUS LOW INITIAL RAS OF A DIVERSE POPULATION

The following graphs show the avg. PRAS effective for various GAs starting at either a low or high PRAS effective population. Crossover only GAs (Figure 41, left) show a clear constraint when starting at a low avg PRAS population, optimising from 21% to up to 58% avg PRAS (UC). These results further demonstrate that UC is better able to optimise when starting at a low level of avg PRAS.

Crossover and mutation GAs (Figure 41, right) show that the addition of mutation enables the GA to improve optimisation for lower levels of initial avg PRAS. Many of the GAs optimise towards an asymptote of +-80% avg PRAS, except for UC,RSAASSEUEM, which is able to optimise towards 100% avg PRAS regardless of the lower initial state.

Higher levels of avg PRAS makes it posssible for all GAs to optimise to 100% avg PRAS, except for the SGA, a key observation. The SGA even decreases in avg PRAS providing a strong indication that there exists a from of counteracting force towards the asymptote of 80% avg PRAS.

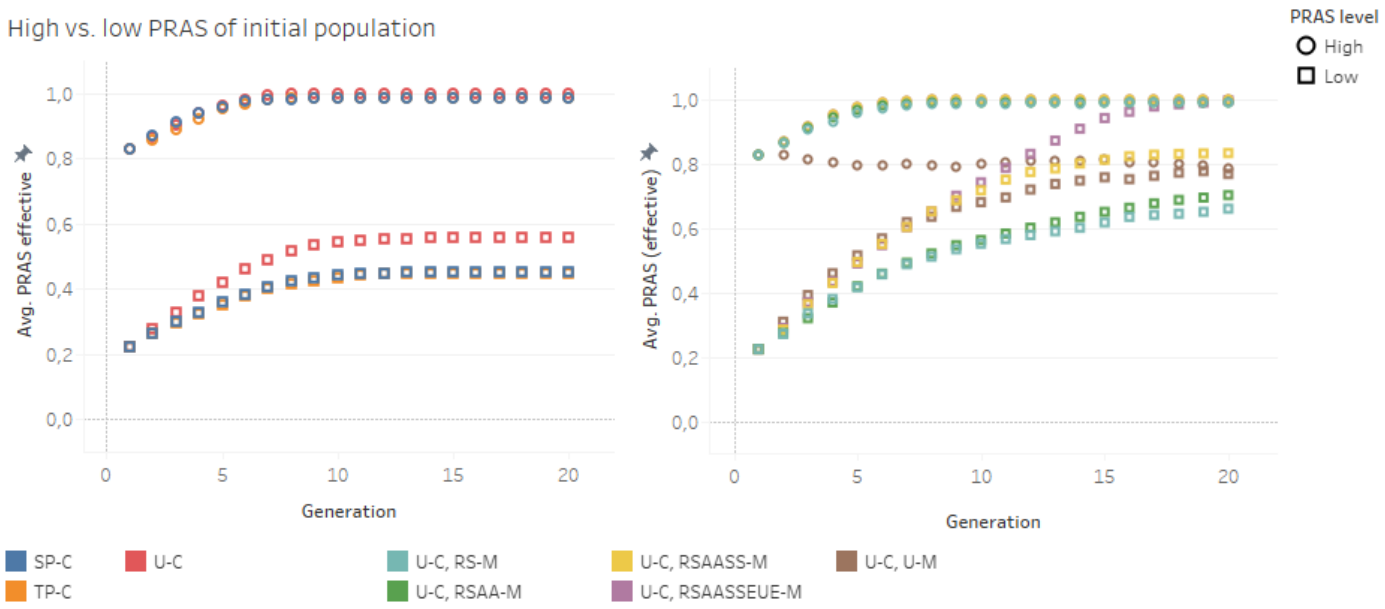


Figure 41 Avg. PRAS per GA for high and low initial PRAS with only crossover (left) and both crossover and mutation (right)

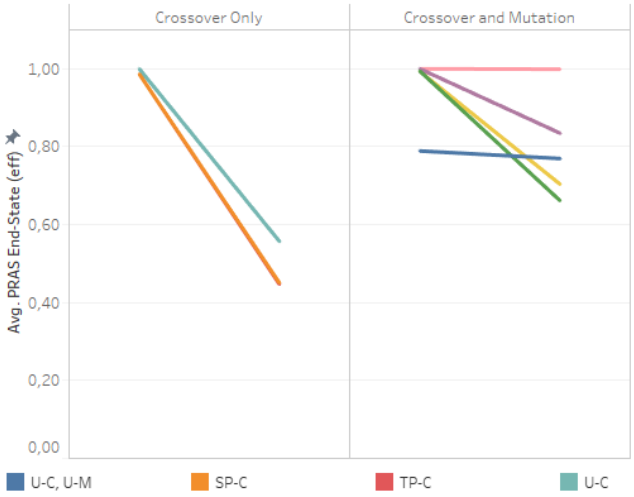
Further investigation of the GA function events demonstrates that UC, UM with a high initial PRAS yields a bias towards introducing of NAs while eliminating Pas (Figure 42). This reflects in a negative GA objective functions score of -1.1, indicating that the GA is doing the opposite of what we intended it to do.

U-C, U-M with Gamma = 15 and Initial PRAS = High

	NA			PA			Grand Total
	A	B	C	C	D	E	
Introduction	157	162	79	2	5	2	407
Progression	434	438	305	143	430	435	2.185
Ubiquity	4	2	8	25	124	164	327
Elimination	4	2	8	22	130	164	330
Suppression	429	446	284	170	470	442	2.241
Extinction	168	176	91	4	5	2	446

Figure 42 Total number of events per GA function, gene section and allele end-state for UC,UM, gamma = 15 and initial PRAS = high

The highest GA objective score is achieved by the UC, RSAASSEUEM combination when starting at low levels of initial PRAS (Figure 43). This GA has high values for the introduction, progression and suppression rates while reaching 100% avg. PRAS after 20 generations.



PRAS level	Heuristics	Experiments	Introduction rate	Progression rate	Suppression rate	Elimination rate	Avg. ES RAS	GA obj. function
High	Crossover Only	TP-C	0,00	0,15	0,15	0,00	0,99	0,1
		U-C	0,00	0,15	0,15	0,00	1,00	0,1
		SP-C (baseline)	0,00	0,15	0,16	0,00	0,99	-0,6
	Crossover and Mutation	U-C, RS-M	0,00	0,13	0,11	0,10	0,99	4,4
		U-C, RSAASSEUE-M	0,00	0,15	0,14	0,00	1,00	0,4
		U-C, RSAASS-M	0,00	0,15	0,14	0,00	1,00	0,4
		U-C, RSAA-M	0,00	0,13	0,15	0,00	0,99	-0,7
		U-C, U-M (baseline)	0,09	0,12	0,12	0,13	0,79	-1,1
Low	Crossover Only	U-C	0,00	0,17	0,11	0,00	0,56	9,2
		SP-C (baseline)	0,00	0,18	0,13	0,00	0,45	2,9
		TP-C	0,00	0,19	0,15	0,00	0,45	2,6
	Crossover and Mutation	U-C, RSAASSEUE-M	0,19	0,20	0,12	0,00	1,00	87,1
		U-C, RSAASS-M	0,15	0,20	0,12	0,09	0,83	40,7
		U-C, RSAA-M	0,10	0,18	0,12	0,08	0,70	19,7
		U-C, RS-M	0,09	0,18	0,12	0,09	0,66	16,0
		U-C, U-M (baseline)	0,13	0,14	0,12	0,12	0,77	9,8

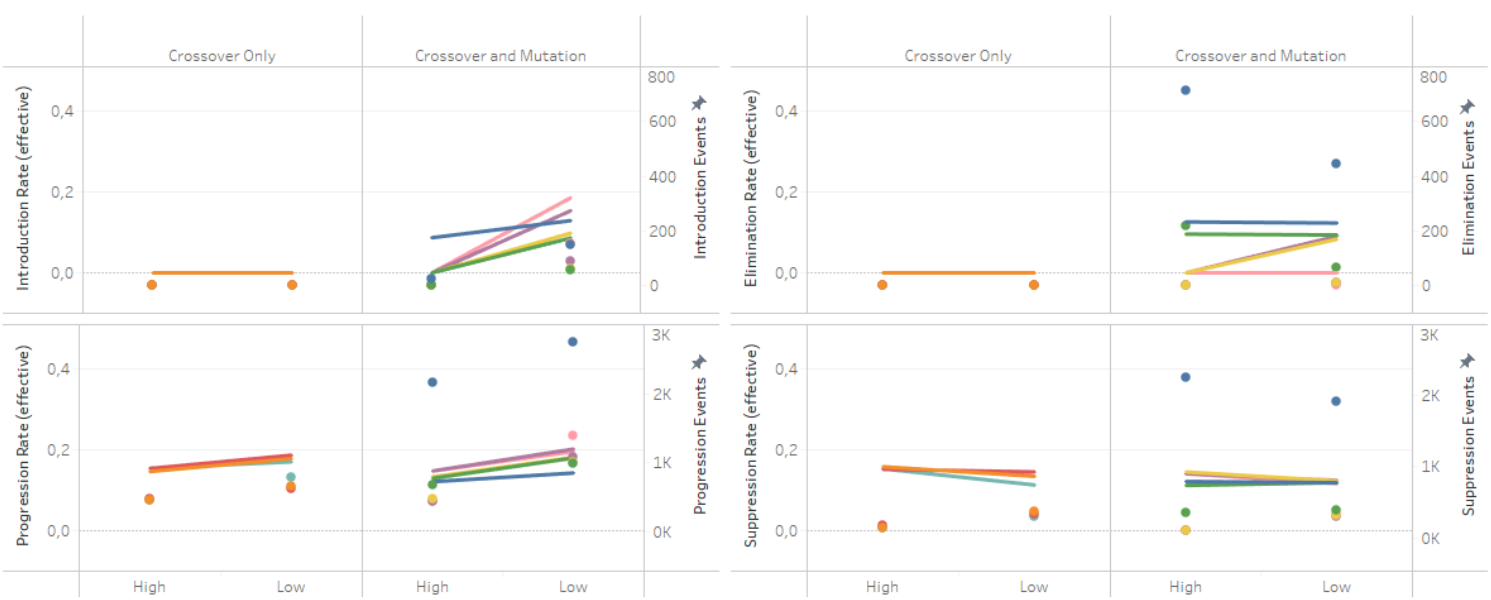


Figure 43 Avg. end-state PRAS for low and high initial strength populations (top left) and their GA function rates with events (top right, bottom)

This IGA has a higher introduction rate as compared to the 2nd scoring IGA, that with UC, RSAASSM. Both progression and suppression rates are identical. Not only is the introduction rate higher, another observation is that this GA has a relative higher number of progression events than all other GAs starting at lower initial PRAS.

The difference in GA function rates between the 2nd best and 3rd best scoring GAs (concerning selection bias) mainly differ on introuction and progression rates.

Further comparison on GA function rates and events in Figure X shows that introduction rates and events increase for lower levels of initial PRAS in the case of crossover and mutation GAs. Furthermore its visible that the SGA with UC,UM has a much higher number of progression, suppression and elimination events than all other GAs.

5.3. PARAMETERS

This section shows the experimental results of a sensitivity study where quantitative parameters such as the mutation rate and problem granularity were varied.

5.3.1. MUTATION RATE

The mutation rate varied from 5% to 20% while the default setting for other experiments in this research was a mutation rate of 10%.

Overall the results demonstrate that the IGAs became independent on the mutation rate while the SGA (baseline) was negatively correlated with an increasing mutation rate. Furthermore, it shows that a lower mutation rate of 5% strongly increased the end-state avg PRAS of the SGA. It seems that the asymptote that often stops the progression of UC, UM shifts up with a lower mutation rate as depicted in Figure 44.

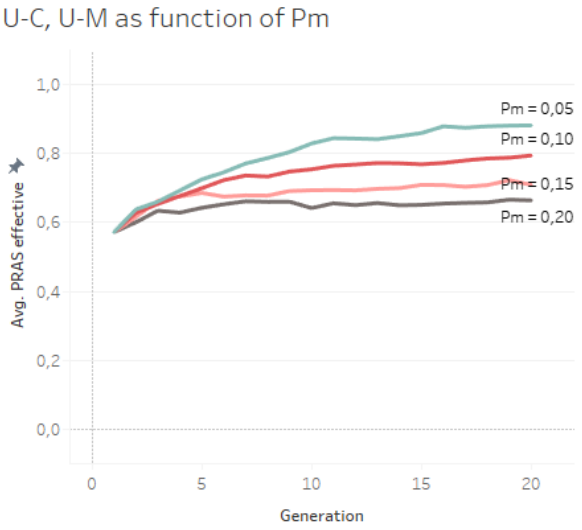


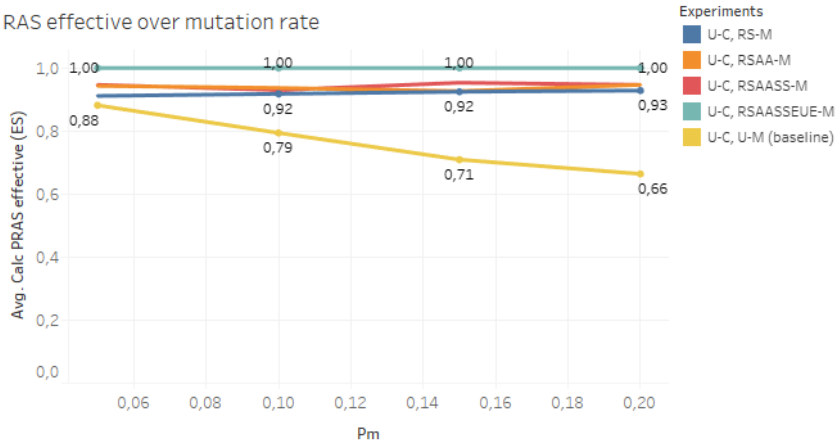
Figure 44 Avg PRAS over generations for UC,UM with varying Pm

The GA function rates and events show that the difference is likely to be driven by a change in the number of progression, suppression and elimination events, rather than changes in the GA function rates.

Furthermore there are fluctuations in the rates for some experiments. In the case of UC, RSM we see that the introduction rate drops to 0 at Pm = 15%, yet for other ranges it remains stable at around 0.1. For UC, RSAASSM the results show a similar behaviour at Pm = 5% and 15% for the elimination rate.

An additional question is whether an increased mutation rate makes mutation heuristics less dependent on the initial strength of the population. **Error! Reference source not found.** provides an overview of the avg e

nd-state PRAS for varying initial population strength (low vs mid) while maintaining a high level of mutation rate (0.2). A primary observation is that lower initial population does not affect UC,RSAASSEUE in reliability as it reaches 93% PRAS with a likely continuation to 100%, yet it does take a higher number of generations to achieve this level of PRAS. This is in contrast to UC,UM, which rapidly optimises to the same PRAS level as when starting with a mid-strength initial population.



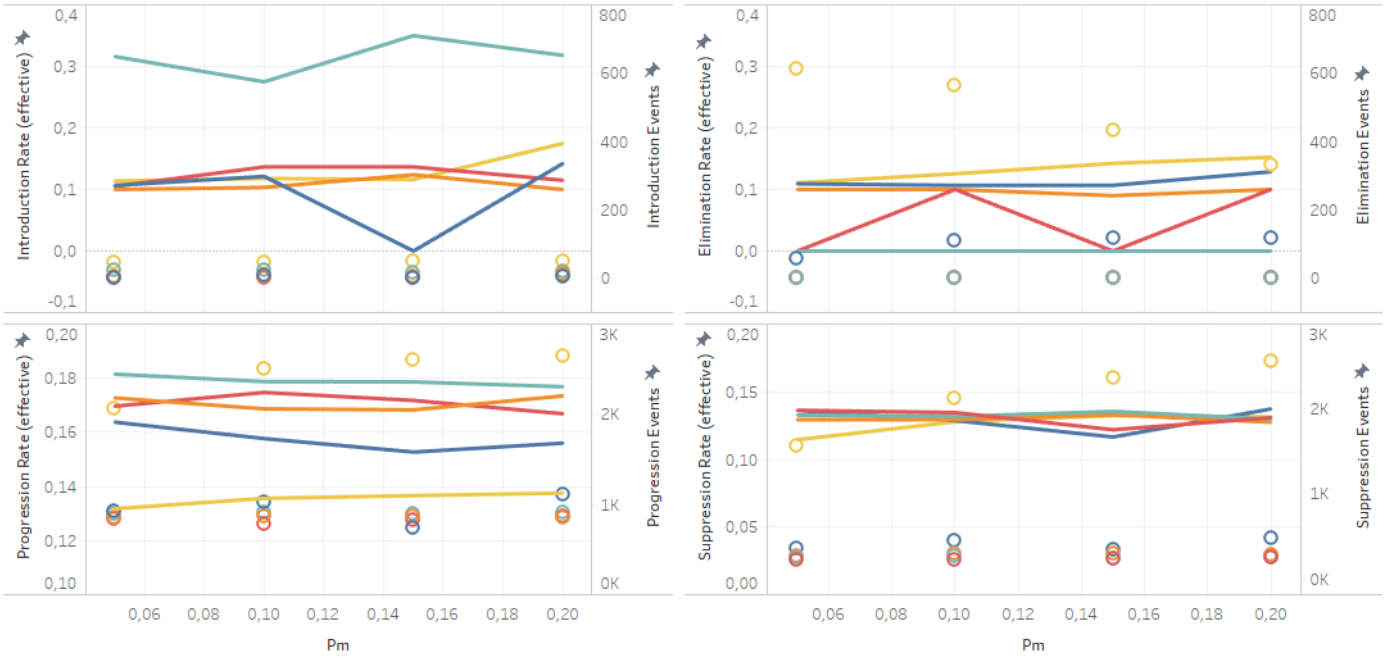
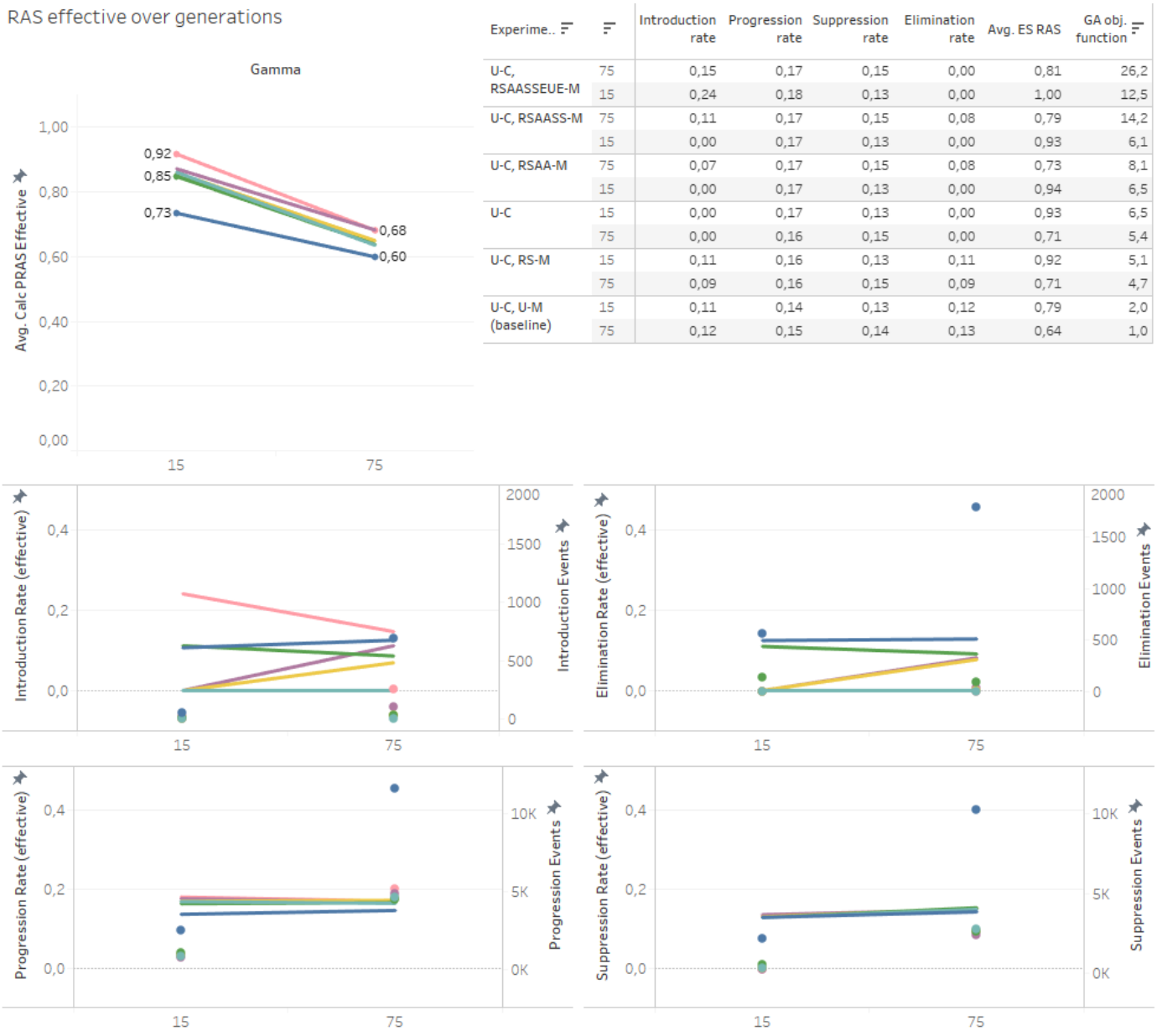


Figure 45 End-state PRAS (top right) and GA functions rates with events (bottom) for a varying mutation rate Pm

5.3.2. GRANULARITY

Granularity determines the number of genes and alleles used in describing the design problem. Increasing the granularity increases the complexity of the problem through a higher degree of freedom for the GA heuristics. Nonetheless, the complexity of the objective function remains the same as the objective function remains the same (F_3).

Results in Figure 46 show that the avg. end-state PRAS negatively correlates with a higher granularity for each experiment group. Higher granularity decreases the introduction rate for UC, RSAASSEUEM while the number of introduction events increases. Furthermore, the number of progression events also increases, yet this does not come with a reduction in progression rate.



Experime..		Introduction rate	Progression rate	Suppression rate	Elimination rate	Avg. ES RAS	GA obj. function
U-C, RSAASSEUE-M	75	0,15	0,17	0,15	0,00	0,81	26,2
	15	0,24	0,18	0,13	0,00	1,00	12,5
U-C, RSAASS-M	75	0,11	0,17	0,15	0,08	0,79	14,2
	15	0,00	0,17	0,13	0,00	0,93	6,1
U-C, RSAA-M	75	0,07	0,17	0,15	0,08	0,73	8,1
	15	0,00	0,17	0,13	0,00	0,94	6,5
U-C	15	0,00	0,17	0,13	0,00	0,93	6,5
	75	0,00	0,16	0,15	0,00	0,71	5,4
U-C, RS-M	15	0,11	0,16	0,13	0,11	0,92	5,1
	75	0,09	0,16	0,15	0,09	0,71	4,7
U-C, U-M (baseline)	15	0,11	0,14	0,13	0,12	0,79	2,0
	75	0,12	0,15	0,14	0,13	0,64	1,0

Figure 46 End-state PRAS (top left) and GA functions rates with events (top right and bottom) for a varying granularity and generations = 20

Figure 47 shows the avg PRAS effective over generations for each of these experiment groups. While experiments with a lower granularity (γ = 15) led to higher end-state PRAS values, these experiments were benefitted due to a higher initial population.

Further extension of the number of the generations for two experiments showed that UC, RSAASSEUEM was able to continue optimisation towards PRAS = 100%, while UC, RSAASSM levelled off at around 81%.

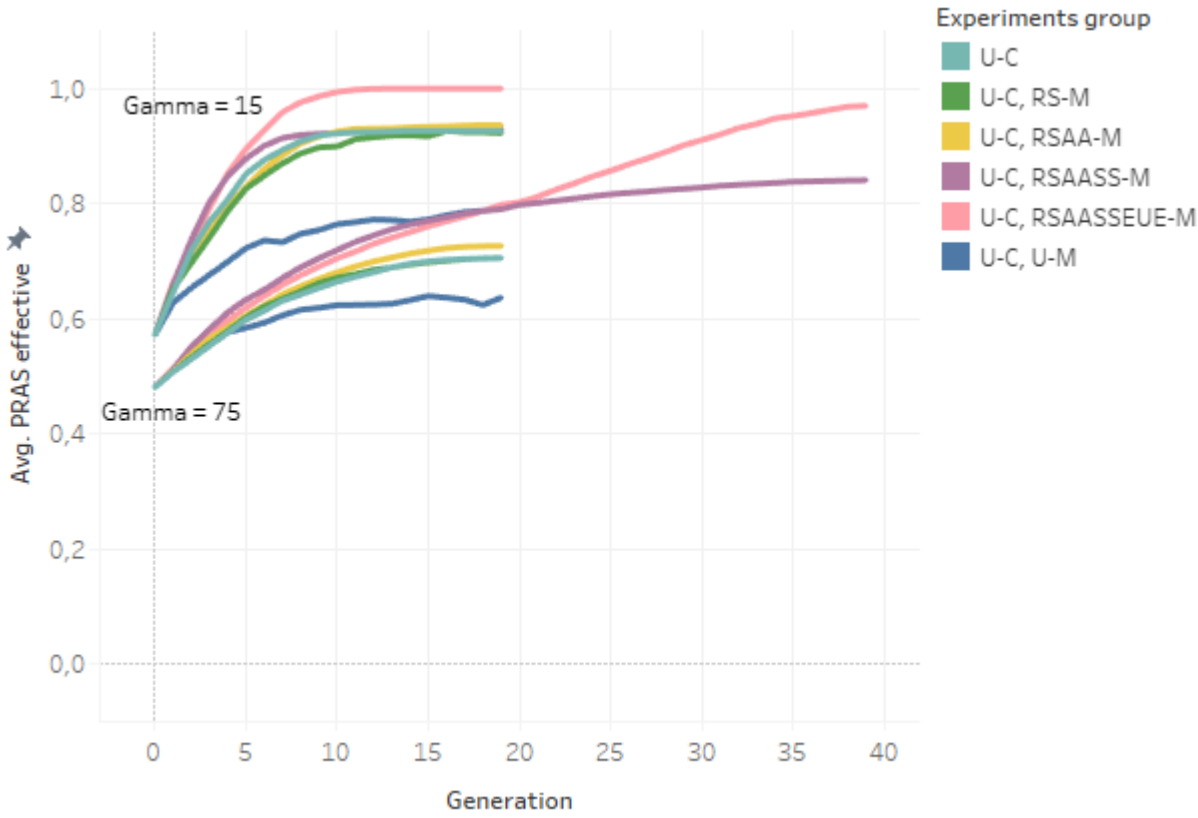


Figure 47 Avg PRAS effective over generations for varying granularity per experiment group

6. DISCUSSION

A literature study on genetic algorithms applied to engineering design revealed that in many cases the choices on genetic algorithm search heuristics is arbitrary. Engineers fall back on the standard genetic algorithm (SGA), typically applied with single point crossover and uniform mutation. These decisions are driven by a lack of understanding of genetic algorithm search heuristics, as well as their interaction with various characteristics of a design problem. Consequently, the results with the SGA have shown themselves to be uncertain, which on its turn has strongly limited the application of genetic algorithms in engineering design (Wolpert & Macready, 1997). This is unfortunate, as capturing the benefits of a manufacturing technique such as additive manufacturing requires a reliable design methodology for highly granular and "organic" components for the inherent combinatorial design complexity. The research goal was to improve our understanding of these search heuristics, both standard and improved, on a simple design case study of thickness distribution. The analysis of the results has brought forward a thesis along with 2 supporting statements.

The following discussion aims to inform practitioners on the nuances when applying GAs to their design problems, as well as bring up more research questions.

The application of the nature-inspired genetic algorithms on damage tolerant fuselage design is flawed and deceptive

Currently, at best, the standard genetic algorithm model is a simplified representation of natural evolution that has proven to be able to optimise some design cases, yet we know very little on when its reliable and how it interacts with specific design problems (Sorensen, et al., 2017).

Overall findings of this case study indicate that the application of SGAs can be ineffective and even deceptive, supported by the fact that none of the experiments with the SGAs were able to find the optimal crenellated plate reliably. On average, the SGA with UC and UM still reached a PRAS end-state of 79%. However, findings on several IGA designs indicate that reliable GA designs can be achieved. The best performing IGA reached an average PRAS end-state between 90% - 100% depending on a range of conditions for the optimisation.

While these high-level results provide us indications of what search heuristics work under these design case conditions, analysis of lower-level results led to two insights, which are each presented in the form of a supporting statement.

The role of crossover and mutation has long been a topic of discussion (Senaratna, 2005). What clearly is not understood, is to what degree crossover and mutation resemble a fundamental truth about genetic algorithms. As part of this fundamental truth it is assumed that any type of pure randomness is sufficient for a reliable genetic algorithm optimisation.

The role of crossover is currently thought to be exploiting the alleles in the current population by exploring stronger combinations of alleles. That of mutation is considered to be exploring the solution space by creating random allele variance in the population. However, the exact mechanisms, or heuristics, through which they fulfil these roles have been widespread and very hard to reproduce (Sorensen, et al., 2017) In this context, pure randomization is randomness not limited or "biased" by any signal, yet that which only relies on a "flip-coin". Evidence of this assumption can be found in the GAs that engineers currently report in literature, as well as in the standard open source code package for GAs which only contains standard mutations using pure randomization (reference).

Statement on SGA: Random mutation is a deceptive heuristic due to its counteracting of a constructive crossover, while crossover is severely limited in its functionality.

The notion that any type of pure randomization is beneficial to the optimisation is flawed. A large share of inexplicability of the SGA optimisation process comes from the uniform mutation heuristic due to its pure randomness. Through experimentation with both *standard* crossover and mutation in the design of flat panels, we found that random mutation is suboptimal and that it counteracts the beneficial GA functions of crossover for mutation rates between 0.05 – 0.2.

Supporting results

Solution quality (PRAS)

Pure randomization on its own, in the form of uniform mutation, did very poorly when used without crossover, leading to a fluctuating average PRAS end-state of 50% (Figure 48). Combining both standard crossover and mutation led to balanced avg PRAS of 79% (Figure 48). Unexpectedly, crossover alone resulted in PRAS 93% (Figure 49) demonstrating that in this case crossover alone would have been better than constructing fitter solutions without mutation. Furthermore, it supports the hypothesis that mutation counteracts the constructive function of crossover.

GA functions

UC alone did very well on progression and suppression leading to an average PRAS end-state of 89% (Table 13). It could not progress any further as crossover on its own could not perform the GA functions of allele introduction or elimination. Further results show that crossover is able to progress PAs more often and better than when combined with uniform mutation.

Initial population

Although the avg. PRAS decreased with a mid- and high-level initial PRAS, at a low-level initial PRAS, the addition of UM strongly increased the ability for the GA to reach higher levels of avg PRAS (78%). UC only resulted in much lower levels (58%) for low-level initial PRAS, making it very constrained to the initial population strength.

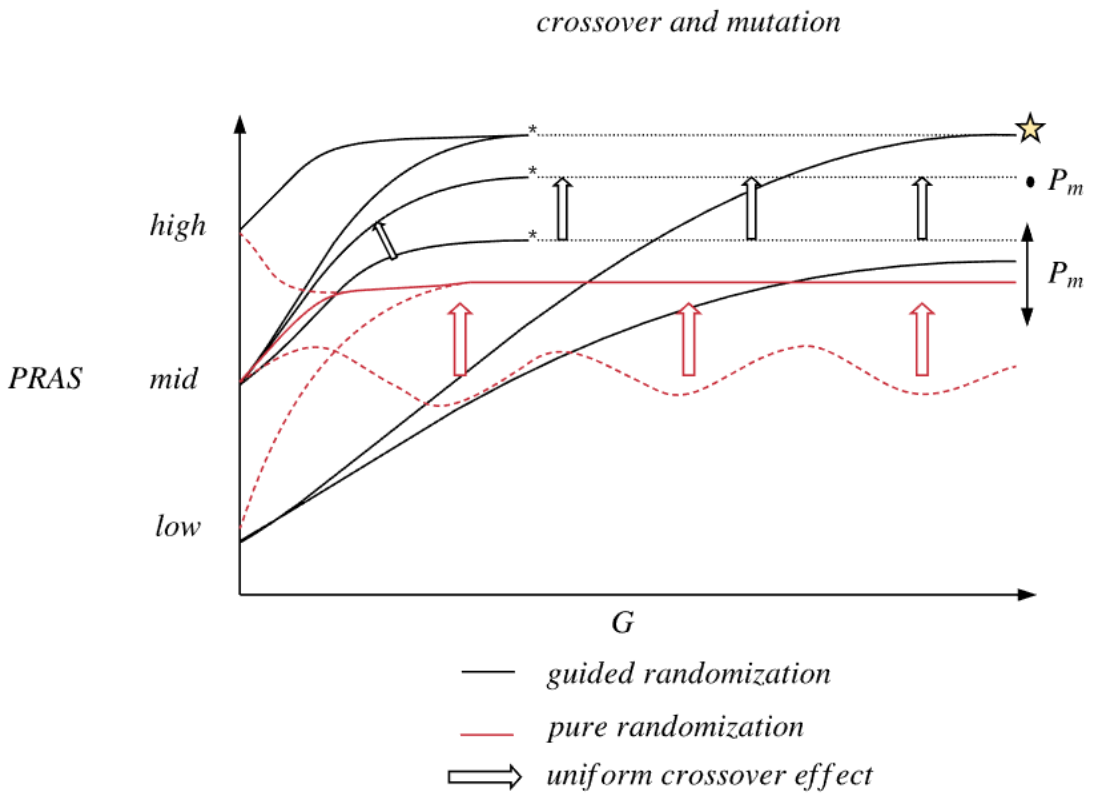


Figure 48 Schematic view of the optimisation process for crossover and mutation SGAs

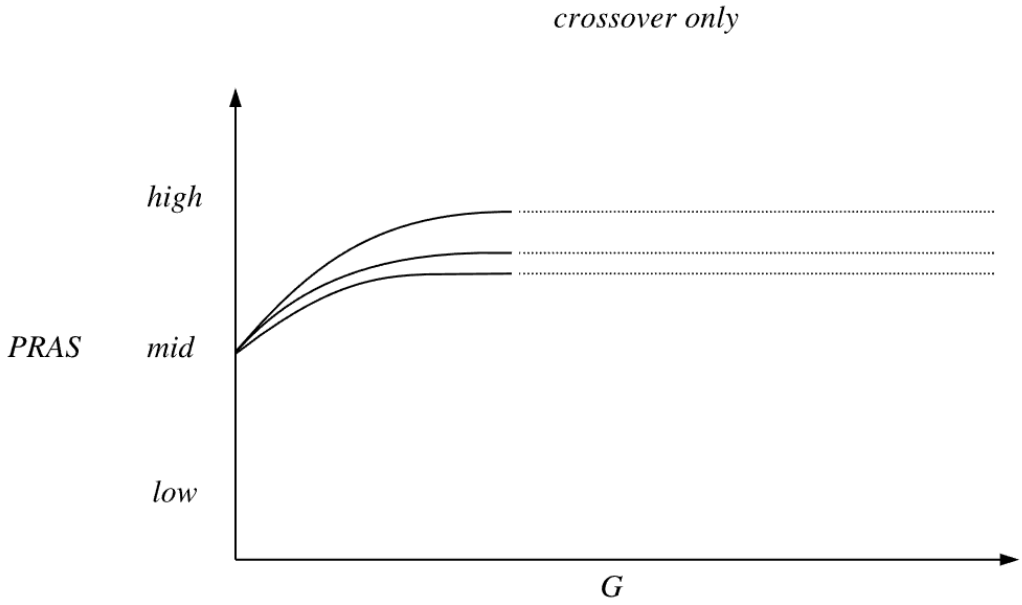


Figure 49 Schematic view of the optimisation process for crossover only SGAs

Mutation rate

Another result supporting that pure randomization can be harmful is that the avg PRAS showed a negative correlation with the mutation rate. Higher mutation rates led to lower avg PRAS levels after 20 generations.

Table 13 Number of GA function events per gene section and end-state grouped by experiment (gamma = 15, F = N/Am)

U-C, U-M (baseline)	Introduction	143	123	45	311	0	14	5	19	330
	Progression	380	413	292	1.085	144	514	491	1.149	2.234
	Ubiquity	5	4	23	32	36	111	138	285	317
	Elimination	4	4	20	28	36	94	123	253	281
	Suppression	494	511	345	1.350	166	437	441	1.044	2.394
	Extinction	158	143	56	357	1	15	5	21	378
U-C	Introduction	0	0	0	0	0	0	0	0	0
	Progression	58	48	41	147	44	199	153	396	543
	Ubiquity	3	7	5	15	19	57	57	133	148
	Elimination	0	0	0	0	0	0	0	0	0
	Suppression	198	141	114	453	16	69	51	136	589
	Extinction	57	53	35	145	1	3	3	7	152
RSAASSEUE-M	Introduction	0	0	0	0	0	27	23	50	50
	Progression	47	45	44	136	39	165	124	328	464
	Ubiquity	11	14	18	43	20	60	60	140	183
	Elimination	11	14	18	43	0	0	0	0	43
	Suppression	180	159	114	453	8	81	48	137	590
	Extinction	60	60	40	160	0	27	23	50	210
U-C, RSAASSEUE-M	Introduction	0	0	0	0	1	3	4	8	8
	Progression	46	53	58	157	41	174	149	364	521
	Ubiquity	1	6	10	17	20	60	60	140	157
	Elimination	1	6	10	17	0	0	0	0	17
	Suppression	202	185	156	543	13	46	42	101	644
	Extinction	60	60	40	160	1	3	4	8	168

This makes sense as a higher mutation rate, in the case of uniform mutation, will only lead to a higher level of disruption of strong solutions. This result implies that the mutation rate for SGAs has been chosen arbitrarily in other engineering research should be considered as irresponsible. While some researchers solve this issue through a sensitivity study, the fact that there are several hyperparameters such as population size and tournament size, makes it a very time consuming effort to find a combination of the best settings for a given application. Add to that the difficulty to explain why a certain combination of settings is optimal, and the application of SGAs on engineering design becomes unrealistic.

Summary

As such, it seems that using UM for the inclusion of introduction and elimination functions came at the cost of interrupting the progression and suppression benefits that crossover has, as crossover alone reaches higher avg PRAS when the initial population is sufficiently strong and diverse. As we cannot ensure nor validate these conditions, we conclude that the application of SGA with random mutation is deceptive.

Possible reasons

A logical reason for the limiting effect of uniform mutation is that it has a higher probability of destructing high fitness solutions than creating them. While there is a probability that UM leads to a positive mutation, the experimental results confirm that its unlikely to benefit the optimisation.

Figure 50 schematically depicts why the probability for UM to destruct becomes higher when solution fitness reaches higher optimality. The values show how many levels of thickness would yield a positive increment towards the optimal solution. As the solution becomes highly optimal, the number of possible thickness levels that yield positive increments decreases.

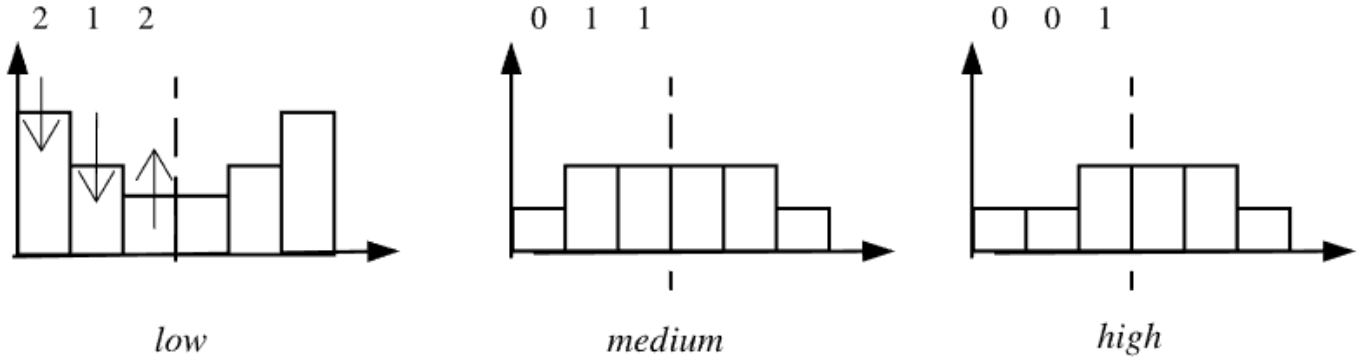


Figure 50 Schematic overview demonstrating that solutions of higher fitness have less potential positive mutations

The counter effect of mutation can be further supported by the fact that we see that the avg PRAS of the initial population only affects the avg end-state PRAS of the SGA to a small degree. We see that in both cases the SGA converges towards an avg end-state PRAS of 79%, even though the high PRAS initial population started on a higher level of avg PRAS (82%). This decline also supports the argument that there exists some counteracting effect, where the destructive effect of uniform mutation increases when the optimality of the solutions in the population increases. Schematically this is shown in Figure 52.

This makes sense, as with weaker initial populations the alleles that are not necessary (NA), or necessary (PA) for the optimal solution are likely (not) present in the population. Standard crossover cannot introduce nor eliminate these alleles and it needs the function of mutation to introduce or eliminate these. This finding is not insignificant, as in real applications one does not know how strong the initial designs really are. If they are initially weak, the GA with random mutation will show strong progress, providing the practitioner with the impression that optimal solutions have been achieved, while in fact, these solutions are still far from optimal (as our SGA experiment with a low initial population reach an average of PRAS = 78%). Furthermore, the output solutions are likely to be different each run and therefore will be considered unreliable.

While one may argue that the disruptive effect of mutation must be considered in relation to selection pressure, since selection pressure doesn't care whether mutation is positive or destructive, it simply needs some variance in the gene pool in order to keep optimising. We don't fully agree with this. Why must a mutation destroy a strong solution in the first place? While selection pressure acts as a solution filtering method, results in this research show that including a mutation filtering, which is in some sense similar to selection pressure, before propagating a mutation can yield better optimisation results. It may be of interest to research how the reliability of the SGA and IGA changes with different selection heuristics and pressures, however we consider it unlikely that changes herein will make the SGA overall more reliable as an optimisation method since the same principles of a disruptive uniform mutation still apply.

While crossover might be able to mix a set of stronger solutions when stronger selection pressure is applied, uniform mutation will become only more disruptive as the number of negative mutations will become even larger when mutating stronger solutions. As such, there exists this counter balancing effect of uniform mutation that stands in the way of finding the most optimal solutions which makes the heuristic deceptive.

Limitations

A critical note on the results is that the number of mutation events was much lower in PRAS compared to UM just by the difference in heuristic design. UM looped through each gene and had a probability Pm that the gene would get mutated, while with IGA, the mutation rate was applied on a solution level rather than gene level. Effectively, this could have meant that the mutation rate on the solution level for UM was much higher than Pm, as the probability that the solution would mutate at least one gene in the SGA would be $P_m * \gamma$, rather than just Pm. Whether this had a significant effect on the results should be considered an important validation to do in further research.

More generally speaking, the limitations of applying the SGA on design cases are numerous. While our results show that crossover through the combination of stronger alleles will counter balance the beforementioned disruptive effect of uniform mutation, our results show that the optimisation process becomes very slow. Even if, in the very unlikely event, it would optimise to highly optimal end-state after a very long time, this would not make it a practical optimisation approach in engineering design or any other application where do not have to liberty, as so to speak, to wait for millions of years as nature would have done. Of course, this SGA does not in any form imitate the complex behaviour found in nature.

Statement on IGA: Heuristic design should be driven by GA functions and knowledge of the fitness space, rather than be based on flawed nature-based models. Crossover is not a fundamental heuristic of the GA, rather it is a type of mutation using guided randomization.

By designing and investigating numerous IGAs based on GA functions - the things the GA is expected to do – we were able to show that an improved mutation can create a reliable optimisation methods despite not including the standard crossover heuristic. Through the construction of a mutation heuristic based on a balanced fulfilment of GA functions, we were able to create a reliable and explainable GA for the identification of damage tolerant crenellated plates in a simplified representation. This balanced fulfilment was achieved by employing forms of guided randomization using concepts as mutation filtering, PRAS and selection bias.

Supporting results

Solution quality (PRAS)

To support these statements we shall discuss a number of insights. First, the best performing IGA was RSAASSEUEM which reliably optimised towards 100% end-state PRAS within 20 generations for all granularities except gamma = 75. In this case of high granularity, the guided mutation was still the only heuristic to continue optimisation towards 98% PRAS. It is assumed that a few more generations would have

led to a reliable 100% optimisation, demonstrating that increased granularity is likely to increase the time needed for optimisation, yet it does not introduce a constraint in the maximum fitness achievable. On its own, crossover performed well with 89% PRAS. The addition of UC with RSAASSEUEM showed an increase in optimisation speed, but it was not necessary in order to create a reliable GA. Figure X and Y summarize our findings to support our statement.

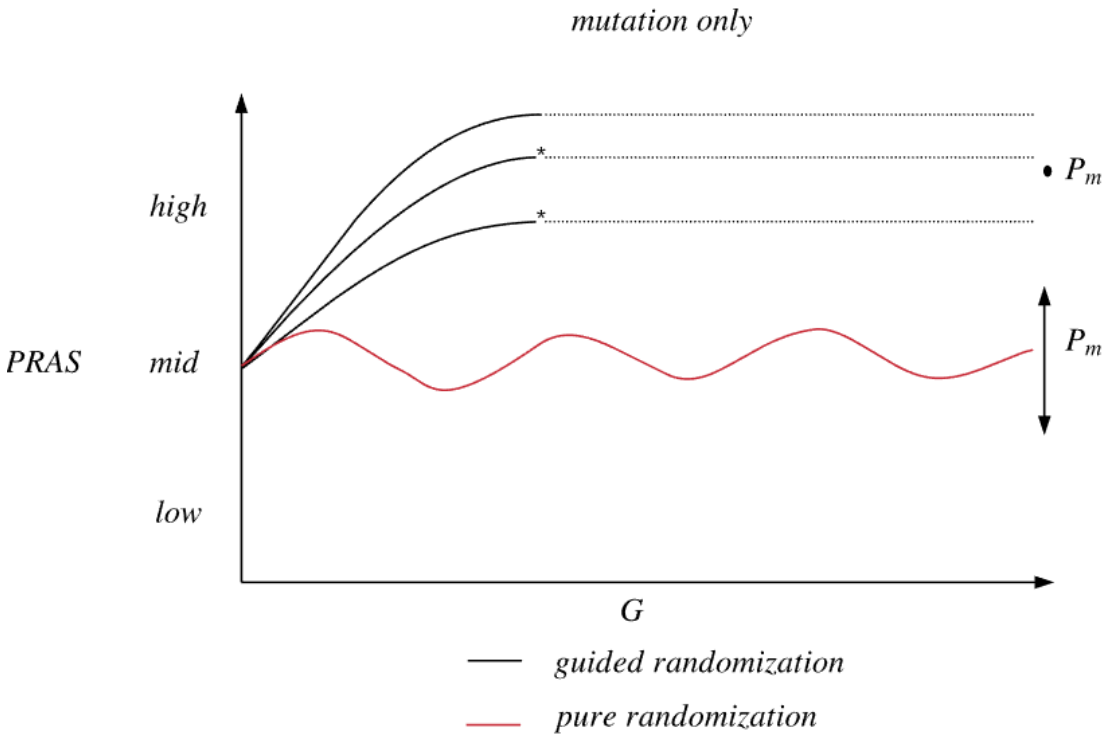


Figure 51 Schematic view of the optimisation process for mutation only GAs using pure or guided randomization

GA functions

More detailed results on the GA function events in Table 13 indicate that guided mutation was able to cover the traditional functions of progression and suppression reserved for crossover, while allowing for introduction and elimination yet reducing the counteracting function of traditional random mutation.

The traditional SGA with UC,UM has an undesired high number of introduction events on NA gene sections A, B and C, as discussed in the previous statement, while RSAASSEUE-M has zero.

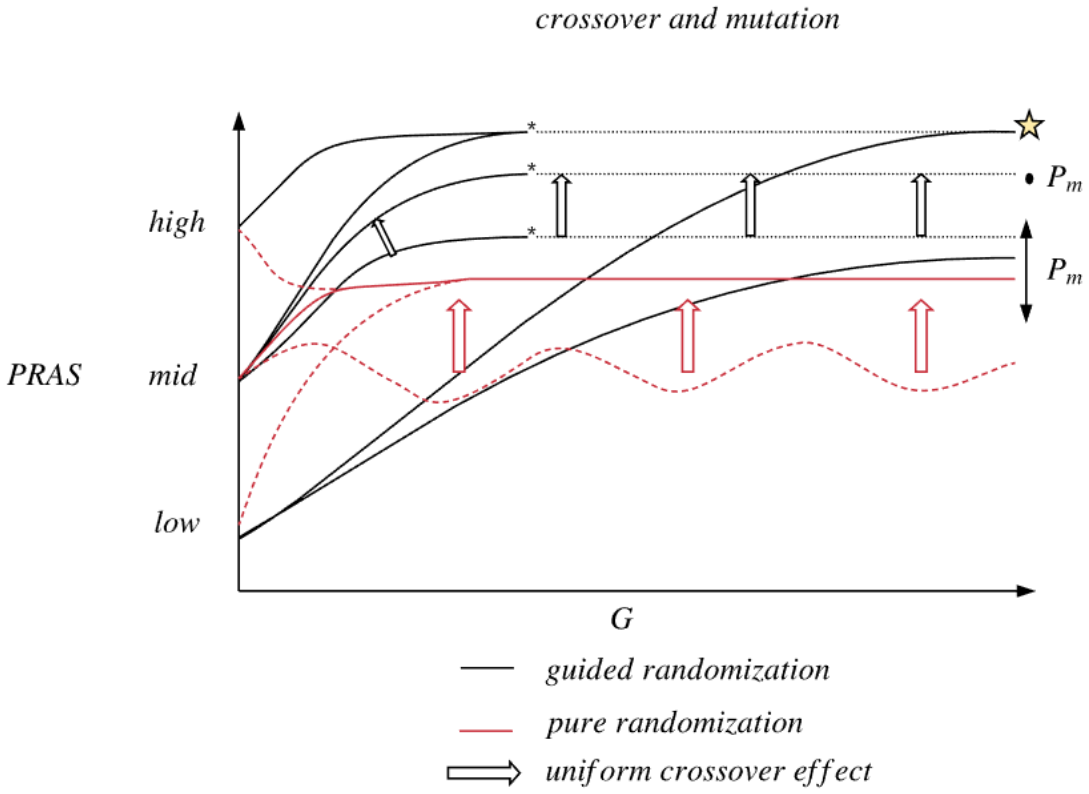


Figure 52 Schematic view of the optimisation process for crossover and mutation using pure or guided randomization

Comparing UC and RSAASSEUE-M on the GA functions of progression and suppression, we can see that the undesired progression of NA decreases from 147 to 136. For progression of Pas, it can be seen that there is a large drop from 396 to 328. The number of suppression of NAs remains equal in total with a slight redistribution per gene section and the undesired suppression of PAs does not change significantly. These results indicate guided mutation was able to fulfill the roles of crossover fairly well, except for the progression of PAs were it lacked, yet in total as it could introduce and eliminate it was a sufficient comprise in being able to reach the optimal solution with PRAS 100%.

If we then compare the addition of UC with RSAASSEUE-M to understand why the addition led to a faster optimization, we see that the number of progression events on PAs increases back to 364, which is closer to the level of UC only with 396 events. Furthermore, the undesired suppression events on PAs makes a large drop from 137 to 101, where the desired suppression of NAs increases strongly from 453 to 543 as well. An undesired effect is that the progression of NAs also increases again from 136 to 157.

Initial population

Lower initial populations led to optimization limitations for most IGAs, while high initial populations would ensure that all IGAs would reach PRAS 100%. Only RSAASSEUEM was unaffected by the initial population, making it a reliable GA.

Granularity

The IGA showed that increased granularity (gamma = 150) did not inhibit it from further optimization as it achieved 98% after 40 generations. The optimization did slow significantly, yet it did not come at the cost of reliability. Furthermore, its likely that a well-informed set of search heuristics becomes more important. UC,RSAASS, while being very similar, levelled out at 83% PRAS. Whether

Mutation rate

Further results showed that all the IGAs became independent of the mutation rate. This is likely to be due to the fact that the mutation rate did not affect the number of mutations on a solution, like standard mutation, but only the probability that a mutation would take place. Furthermore, in the IGA each mutation was ensured to be beneficial, or it wouldn't be propagated. Thus, increasing the mutation rate would just increase the number of not-propagated mutation events if only bad mutation would be left.

Possible reasons

How can these results support our understanding of crossover and mutation? If these heuristics are supposedly different elements of a GA, then why does RSAASSEUEM yield reliable high levels of PRAS without UC, which is considered a critical component of the GA?

Further investigation through the perspective of GA functions shows that crossover and guided mutation are not that different. First, the signalling of allele strength. RSAASSEUE was able to "select" alleles likely to yield positive increments in fitness by using PRAS as a signal in mutation. PRAS was constructed by relating the fitness of solutions in a population to the occurrence of certain alleles.

Crossover traditionally does the same, but then through parent selection. When two parents get selected in tournament selection, implicitly we are estimating the strength contribution of their specific alleles in relation to their superior fitness in the population. Crossover then uses the set of alleles in the parent solutions to "mutate" both parents into different solutions yet with the same set of alleles. In this sense, the candidate list of alleles with highest PRAS is also a set of potentially strong alleles. The difference is then that crossover samples entire solutions, while RSAASSEUEM samples specific alleles.

Just like with RSAASSEUEM, alleles in high fitness parents have a higher likelihood of being mutated towards in others solutions when using uniform crossover. *Therefore we derive that both methods utilize implicit ways of signalling which alleles are likely to yield fitter solutions through some modification.* Therefore both methods attempt to fulfil the GA functions of progression and suppression optimally.

Another similarity is the selection bias used in RSAASSEUEM. Crossover employs a similar bias as the selection of high fitness solutions through a tournament selection. This means that on average the stronger solutions will produce more offspring solutions and share their alleles. In RSAASSEUEM the same happens through the selection bias we introduced, as the strong solutions are more likely to get a positive mutation and thus be more likely to produce offspring in the next generation due to their increased fitness.

Both heuristics also employ a mutation filtering strategy. Where RSAASSEUEM does so explicitly by evaluating the fitness increment of a potential mutation, crossover applies several mutations and is dependent on the selection, colloquially referred to as survival of the fittest, to cumulatively evaluate and propagate beneficial mutations in the population. Nonetheless, the outcome of both heuristics is driven in the same direction – progression or suppression of the right alleles through filtering of beneficial mutations. Yet, the key difference is that crossover does this cumulatively through selection, while RSAASSEUEM does this on a single mutation event basis.

A key difference between RSAASSEUEM is the number of changes a single mutation event can apply to a given solution. In the case of crossover, a solution is likely to be changed on multiple genes. The number of genes that are mutated during each crossover event differs since UC considers the possible mutation of each gene separately. In contrast to RSAASSEUEM, which only applies a single mutated gene when a mutation event takes place. This could explain why the addition of UC with RSAASSEUEM still yields an increase in optimisation speed, as crossover increases the number of positive mutations. Yet, it might also introduce some negative mutations, however our results indicate that overall this might not be harmful to the optimisation.

Another key difference is that crossover is still more likely to yield a negative mutation than RSAASSEUEM, which may explain why RSAASSEUE-M also reduced the number of progression events on NAs in comparison with UC (Table 13). While both heuristics employ a mutation filtering strategy, crossover does so implicitly through the use of survival of the fittest, while RSAASSEUEM does so explicitly by evaluating the increment of fitness per allele during a mutation event. This delay in evaluation in crossover still makes it possible that negative mutations are applied to strong solutions. An extreme example would be when crossover yields two offspring solutions that are both less fit than their parents – which is possible.

The expected probability that RSAASSEUEM yields a negative mutation would be considered small and this would be dependent on the spread in fitness increments that a given NA could have. A given allele does not have a fixed fitness contribution, rather the fitness contribution is dependent on the solution that it is part of. As such, if certain alleles have a high spread in fitness increments depending on the solution context, and if part of those increments are positive while the allele is not part of the optimal solution, then RSAASSEUEM will propagate that allele. Further research is would be necessary to precisely determine the implications of such negative mutations on the optimisation process.

We conclude from our results that its more important to have a heuristic, or set of heuristics, that together fulfil the GA functions, rather than limiting ourselves to the standard definitions of crossover and mutation inspired by nature. Whether we call them crossover or mutation in the first place then becomes irrelevant.

The same controversial conclusion that crossover is a form of mutation was made by David Fogel, as documented in a summary on a debate between the roles of crossover and mutation by (Senaratna, 2005).

Limitations

So if guided mutation is a better functioning heuristic based on GA functions, what are some important limitations of this heuristic? Under which condition must an engineering designer be cautious in applying these genetic algorithms? A number of limitations is discussed grouped by specific topics of interest to engineering design.

The importance of understanding the fitness space

While experimental results with RSAASSEUEM were promising as well as insightful to better understanding important mechanisms of the GA, this heuristic is expected to have limitations.

The first limitation would be when a solution space is largely flat in terms of potential positive mutations in the fitness space – called plateaus (Fonseca & Fleming, 1995). These plateaus are problematic as the changes in fitness are very small, even near to zero. This can make the optimisation directionless, leading to a state of genetic "drift" where the population changes in terms of genotypes, yet not in fitness (Fonseca & Fleming, 1995). In this case of RSAASSEUM, which would require positive fitness increments to guide its mutation, this would make it default back to the zero frequency alleles which is likely to lead to genetic drifting or the absence of any mutation.

Furthermore, this heuristic would probably not work on rugged plateaus, meaning that to cross such a plateau, you would need many small positive, but also small negative changes. Since this heuristic only accepts positive changes, it would not allow to traverse a lightly rugged plateau (Weise, et al., 2009).

Additionally, this mutation is likely to fail in problems where only the combination of certain design characteristics unlocks fitness increases, while all intermediate solutions do not. This concept was coined the Royal Road Function by Mitchell et al. (Mitchell, et al., 1991). Figure 53 shows such an example with binary values where only the combination of 4 alleles in a single block unlocks a large jump in fitness. Such a fitness landscape would require the heuristic to contain some manner of mutation blocks of genes, else it would be impossible to arrive at such solution. Currently, the RSAASSEUEM heuristic treats alleles separately and it would not get any reinforcing "signal" to build this block. However, the SGA would be expected to have similar issues, as even with random changes the intermediate solutions would have to survive selection, or

fitness	$F = 10$	$F = 10$	$F = 100$
genotypes	0000 0000 0000	0000 0001 0000 0000 0010 0000 0000 0100 0000 0000 1000 0000	0000 1111 0000

Figure 53 Schematic overview demonstrating how a "building" blocks of alleles unlocks fitness while single alleles don't

the small chance that these 4 are combined into the last solution would have to occur.

Furthermore, it is confusing to mention that GAs are proficient in creating building blocks, when in fact not all design problems require the formation of building blocks to reach the optimal solution, as seen in this

research. While SGA must be proficient in optimising problems with building blocks, where our IGA might not, it does not mean that the SGA will do well in problems where building blocks are not required. This again shows how important the understanding of the fitness space is in order to correctly identified the heuristic requirements. We are not sure if this led to the inappropriate application of SGA as seen in literature, yet the research did not indicate any thorough investigate into the characteristics of the fitness space, thus it seems plausible. In this research they demonstrate that crossover is essential on Royal Road functions (Jansen & Wegener, 2005), yet it's likely that adapting RSAASSEUEM to handle blocks of alleles rather than single alleles would yield similar, or even more reliable, results than single or two point crossover. As such, we argue to it doesn't prove that crossover is essential, rather that understanding of the fitness space is essential to design a proper set of heuristics.

As these examples illustrate, knowledge of the fitness space is crucial in order to understand how certain heuristics will interact on it. Even though evolutionary algorithms, including genetic algorithms, have shown potential in handling complex fitness spaces including discontinuities, multimodality, disjoint feasible spaces and noisy functions, the results of the SGA on the case study show what limitations exist when heuristic design is not informed by knowledge on fitness space characteristics. This research has not studied the fitness space of the solution space in detail. It would be of interest to determine how the results of this research were influenced by the characteristics of the fitness space.

Engineering design requires solving complex fitness spaces

It is known that the fitness space in this research is rather simplified. While two competing objectives (weight versus damage tolerance) did exist, it is realistic to expect several additional objectives such as stiffness and buckling resistance in the engineering design of fuselage plates, as well as having more complex models which more accurately capture the real behaviour.

Often, these multi-objective optimisations yield more complex fitness spaces (Fonseca & Fleming, 1995). Through the application of several engineering models, one effectively increases the complexity of the fitness value and thus fitness space.

First, these fitness spaces are more likely to contain more hard trade-offs as increasing number of competing objectives are introduced. In this case study, we had a single optimal solution. However, it is likely that in these cases there exists several equal optimal solutions (= identical fitness values), with some more optimised for certain objectives than others.

Solution fitness has an important role in guiding the optimisation process of a GA. A key issue with the fitness score is that GAs require a single scalar value upon which selection through "survival of the fittest" occurs. In practice, objective functions are thus combined and aggregated into a scalar function according to some understanding of the problem. However, as the unexpected results of the trivial, full block solution in this research demonstrate, capturing the true intention of a designer is not trivial.

Additionally, creating an objective function that balances many objectives is likely to lead to poor performing solutions on most objectives, rather than more optimal solutions for less objective functions. If the designer

does not know how much has been sacrificed for the satisfaction of another objective function, it will be more difficult to make a good trade-offs. As such, it may be insightful to try several, extreme, objective function weights in order to find out what the designer truly cares about, something that is unlikely to be clear a priori of the optimisation.

Setting the weights for in competing objectives is likely to remain an activity for the designer. Due to its nature it does not allow for a single, perfect solution, but rather a grouping of perfect solutions, often referred to as the Pareto-optimal set (Fonseca & Fleming, 1995), based on what value the designer places on some trade-offs. In theory, the GA functions in this research would remain the same, namely for the given objective function to find progress PAs and suppress NAs. What changes is the distribution of fitness increments that an allele has given all possible solution contexts.

However, we may assert that knowledge of engineering models will provide benefits in designing appropriate heuristics for the genetic algorithm. For instance, one could learn about the structure of the fitness space through more detailed understanding of the interaction of various mathematical models. This knowledge could inform the decision on the proper heuristic design, say if the fitness space is mainly characterized by fitness blocks, as discussed previously. In the context of this research, without further investigation in the physical models, it is very hard to tell what effect these models have had on the results of this research. Further research would be of interest to understand this better.

It is likely that RSAASSEUEM will still face challenges in optimisation in such complex functions. First, the IGA will continue to work with the fitness value as described in the previous parts of this thesis. Alleles that are beneficial in terms of several models will likely receive higher fitness increments and thus quickly propagate within the population using RSAASSEUEM, compared to those alleles which are beneficial in less models.

Third, the risk of a path dependency in multi-objective optimisation is likely to become larger with increasing number of models. As the optimisation progresses in a certain direction, certain PAs will consistently yield negative fitness increments, even though they might have been part of a solution contained in the optimal solution set. These alleles would have yielded positive increments if the population of solutions had evolved in a different direction. To understand in what degree this inhibits a reliable GA, further research is necessary. It is expected that RSAASSEUEM will have difficult in such an optimisation, as this set of heuristics is strongly oriented to fitness increments and does not permit very random moves that could that it out of a certain path.

Fourth, any limitations or errors in any (engineering) models are likely to be extrapolated in the objective function and thus influence the optimisation process. This concerns a general issue with GAs, as these algorithms are not able to evaluate errors. Increasing the number of (imperfect) models increases the number of errors when assumptions and limitations are stacked. It would of interest for further research to knowingly introduce certain errors into the optimisation and track to what extent these influence the optimisation process. To the best of our knowledge no research has been published about this.

Finally, if there is very limited knowledge of the fitness landscape, more implicit strategies might still work. For instance, engineering designers might test various types of heuristics specifically designed to work well

with certain fitness landscape characteristics to derive what characteristics are likely to be present in the fitness space. This would increase the confidence of certain designs over time.

These examples provide evidence of the importance in understanding the general features of a fitness space to understand what heuristics are applicable. Some research directions in defining features of the fitness space are the development of statistical measures to determine the presence of features of interest or understanding the correlation structure of fitness spaces (Forrest & Mitchell, 1993)

Higher granularity engineering problems

In engineering design, granularity will be higher than presented in the case study of this research. Every additional gene increases the length of a genotype and thus the degree of freedoms. A higher number degrees of freedom is another manner to increase the complexity. Also, granularity may be increased by introducing several material options, more detailed shapes and hierarchy of designs. Similar to DNA, these concepts must be encapsulated in a genotype for the mechanisms of heuristics search to work on.

Furthermore, in engineering there are often various levels of designing features. Often, design starts on the higher level and increasingly designs smaller features. In the case of the GA, it would struggle with designing on different levels. While it may be suitable to sub design the configuration of a certain fuselage plate, as done in this research, incorporating this design in a higher level, say the entire fuselage, would at this point not be feasible. For this, the potential solutions must be configurable in a single genotype.

The question is what type of granularity is required for finding more optimal solutions. Granularity as we investigated in this research makes it more difficult for the GA to identify the optimal solution, yet since we know the optimal solution, we know that it hardly brought any additional fitness. This indicates that we must consider what type of granularity is likely to yield larger jumps in solution fitness. As such, a progressively refined search employed by (Lu, et al., 2016) seems to be a proper approach to only adding granularity if beneficial. A more interesting direction of research into granularity would be to investigate the effect of adding different material options.

However another issue is that all constraints need to be defined in the genotype, or through some genotype-phenotype mapping. However a genotype-phenotype mapping that causes unfeasible solutions to map to the same phenotype would cause an increase in genetic drift, hence causing both the SGA and RSAASSEUEM to receive poor signals for improving the genotype.

It is not possible to say whether the same levels of avg. PRAS would have been reached if the initial population would have been on the same level. Nonetheless, results for these experiments when providing 40 generations demonstrate that RSAASS runs into a limit, while RSAASSEUEM is able to continue optimising towards 100% PRAS.

The heuristic components necessary for effective optimisation

While the better understanding of RSAASSEUEM provides inspiration for further experimentation, designing such heuristics was not without any risks. Many improved heuristic designs turned out to be undesirable

after investigation of the experimental results. As such, these results demonstrate that the design of heuristics requires careful consideration and that the perspective of GA functions can provide guidelines. While this research had the benefit of knowing the optimal solutions in order to benchmark the heuristics, in real life applications this luxury does not exist. This warrants a critical discussion on the heuristics evolving around the question whether each component of the IGA is really necessary.

While RS-M was able to propagate PAs, the results show that its limitation was that it kept introducing NAs into the population. This is likely to happen when a large share of the alleles have similar PRAS values when a population has converged. In that case, the algorithm will get a list of potential alleles with identical scores to mutate towards. By definition that the population has converged, the mutation will reach the fallback and introduce a zero frequency allele. If the solution has converged on a level of high fitness, the probability that the introduced allele is a NA is high, thus potentially resulting in a stagnant optimisation. As such, the fallback of the mutation heuristic became a limitation on its effectives in directing the mutation.

The final limitation that each of the mutation heuristics except for RSAASSEUEM shared was the inability to eliminate ubiquitous NAs. Initially, the concept of optimisation and literature was focused on ensuring that alleles which could improve the fitness could be identified and "built" onto the solutions. However, it became clear that it would be just as important to remove bad parts of the designs as well in order to reach the optimal designs. While this can be viewed as anecdotal to this research, it provides an important message that engineering designers can be biased in how to design custom GAs while overlooking key aspects in the design of an effective GA.

One may wonder whether the assumption that the traditional selection heuristic is a necessary component in the GA given the design RSAASSEUEM, as this heuristic was designed to avoid negative mutations, a filtering function that traditionally happens at the moment when selection filters out these bad moves by sub selecting the population based on relative fitness. Further research would be necessary to investigate the effects of removing the traditional selection heuristic from the RSAASSEUEM on the optimisation process. There exists a possibility that the function of selection is also included in the RSAASSEUEM.

Methodology of this research

While it may seem that the RSAASSEUEM heuristic is more efficient due to a lower number of generations it takes to reach a level of PRAS, efficiency should ideally be measured in terms of how often a solution evaluation takes place. Especially in engineering design since an evaluation often times requires most computational effort. In retrospect, using a definition of number of solution evaluations would be a more suitable axis to use as some researchers have done (Forrest & Mitchell, 1993).

Furthermore, in order to fully understand the contribution of each component (mutation filtering, selection bias and PRAS signalling) in the reliability of the optimisation process of RSAASSEUEM, further research is necessary which would include cross experiments with just AASS or RSSS and include the EUE in all other experiments.

In contrast to RSAASSEUEM, crossover does allow for the small probability that a weaker solution is selected as a parent. Also, crossover does perform strongly on the GA functions of progression and suppression. Does this ability in allowing some negative combinations to more effectively sort alleles? It would be of interest to investigate what the importance is of allowing some *negative* moves at points in the optimisation process. Currently, RSAASSEUEM does not allow any negative moves and it should be evaluated whether there exists a specific contribution of these *negative* moves.

Concluding remarks

Possible reasons why we have always thought that crossover and mutation were distinct, is because of our overly dependence on inspiration from nature when it comes to genetic algorithms. Results of this research have demonstrated that the SGA has strong limitations while a more scientific approach to GA design can lead to more reliable optimisation methods.

The discussion further argued that the design of heuristics should be driven by the requirement of fulfilling all GA functions, while letting the characteristics of the fitness landscape determine the precise mechanisms of these heuristics. This is in stark contrast with what we have seen in the literature, where SGAs are applied without critical evaluation of their applicability to a given problem.

As such, this discussion ends with the statement that nature-inspired model of the SGA is flawed and can be deceptive in regards to the optimisation that its presents.

While further research is a clear requirement, the above discussion of the results in our eyes provides enough evidence to warrant a more cautionary approach when applying GAs to engineering design as compared to what was found in the literature.

7. CONCLUSIONS & FURTHER RESEARCH

7.1. CONCLUSIONS

These conclusions have been made within the scope of this research. Thus, further research is necessary to support whether these conclusions hold on other design cases than the crenellated plate for fatigue damage tolerance, as the fitness space would different.

1. **What is the relative importance of crossover and mutation heuristics in the optimisation process of a genetic algorithm designed for damage tolerance of flat panels?**

   1.1. Are there specific functions in the optimisation process of a GA which need to be fulfilled in this application?

   1.1.1. Our hypothesis is that specific GA functions can be defined in the optimisation process for damage tolerant flat panels

   1.1.2. Our conclusion is that certain GA functions of allele introduction, progression, suppression and elimination correlate with improved optimisation results for damage tolerant plates. Designing heuristics based on GA functions allows for better understanding and explainability of the genetic algorithm.

   1.2. How well do the standard crossover and mutation heuristics fulfil these functions within this context?

   1.2.1. Our hypothesis is that standard heuristics are not well suited for their intended functions in designed for damage tolerant flat panels

   1.2.2. Our conclusion is that both heuristics are not well suited for the identified GA functions in the context of designing for damage tolerance flat plates. Mutation counteracts the ability for crossover to progress and suppress the right alleles, while crossover is unable to introduce nor eliminate certain alleles.

   1.3. Could a less random mutation heuristic increase the reliability of the GA in identifying optimal design solutions for damage tolerant flat panels?

   1.3.1. Our hypothesis is that a mutation heuristic which utilizes information about previous searches leads to more reliable outcomes

   1.3.2. Our conclusion is that the addition of PRAS signalling, filtering of mutation and a selection bias provides a reliable GA in the case of designing optimal crenellated plates

   1.4. Under what conditions, if any, do we need both standard crossover and mutation heuristics in a GA designed for identifying optimally damage tolerant flat panels?

1.4.1. Our hypothesis is that in any case crossover is not a necessary component in GA optimisation for damage tolerant flat panels

1.4.2. Our conclusion is that crossover is a type of mutation in terms of the GA functions and therefore does not constitute a necessary component for the GA. In order to create more reliable GAs, it's more important design heuristics which fulfil each GA function to some degree.

2. To what extent do quantitative parameters of the GA influence the optimisation process when designing for damage tolerant flat panels?

2.1. What correlations are there between the granularity and the reliability of a GA for varying values of the population size?

2.1.1. Our hypothesis is that a larger population size leads to higher reliability of the GA in finding optimal results

2.1.2. Our conclusion is that a higher granularity negatively correlates with the reliability of SGAs, while it does not affect the reliability of a GA utilizing PRAS signalling, mutation filtering and a selection bias within the range $\gamma <= 75$. Optimisation does require more generations with increased granularity.

2.2. What correlations are there between the mutation rate and the reliability of a GA for varying values of the mutation rate?

2.2.1. Our hypothesis is that a higher mutation rate will not always lead to higher reliability of the GA in finding optimal results

2.2.2. Our conclusion is that the SGA is negative correlated with the mutation rate as random mutation increasingly destructs the constructing of higher order and more optimal solutions. The IGAs become independent of the mutation rate due to signaling of PRAS.

2.3. What correlations are there between the initial population strength and the reliability of a GA for varying values of the population size?

2.3.1. Our hypothesis is that a higher mutation rate will not always lead to higher reliability of the GA in finding optimal results

2.3.2. Our conclusion is that Initial population has an effect on the time to optimal convergence but not on the reliability of the IGA

3. Can a simple objective function be formulated that captures the intentions of a designer a priori?

3.1.1. Our hypotheses is that simple objective functions will a priori lead to trivial solutions which the designer did not in tend to achieve

3.1.2. Our conclusion is that a simple objective function is not likely to capture the true intention of an engineering designer and that

7.2. RECOMMENDATIONS FOR FUTURE RESEARCH

This section highlights the suggestions for further research grouped by each component of the solution framework shown in the methods section.

- **Engineering models**
    - Determining methods to characterize fitness space elements produced by engineering models, such that this information can inform the design or applicability of certain heuristics. If no proper ways of determining characteristics of the fitness space exist, the design of heuristics is severely limited to trial and error methods in finding effective GAs, which is likely to limit a wider application. This statement is somewhat contradicting to make, as a full characterisation of the fitness landscape would imply that no optimisation will be necessary since the optimal solution can already be identified.

    - Studying the propagation of known engineering model errors through the GA optimisation process with either one of multiple models. Understanding the propagation could prevent engineering designers in getting misled by the GA results.

    - Dynamic objective functions which can be used to steer search in a given direction, find a set of more optimal solutions, and then steer iteratively towards another objective as a way to avoid suboptimal solutions and provide designers several optimal designs based on the weights they provide to certain trade-offs (e.g. weight vs damage tolerance).

- **GA heuristics**
    - Research the effect of changing several alleles in a single mutation event to understand whether it can make the search process more efficient. UC was argued to faster than RSSAASSEUEM because it can process multiple changes in a single mutation event. Adding this capability to RSAASSEUEM is expected to make it more efficient. This could be especially relevant for optimising higher granularity and complexity engineering problems.

    - Research the effect of processing blocks of alleles in as a single mutation and whether this enables the heuristic to be more applicable to Royal Road type of functions. Understanding whether the heuristics are more customizable will provide confidence on how to design heuristics for solving certain characteristics of the fitness space.

    - Research more advanced ways of implicitly communicating allele strength, such as maintaining historic view of (relative) allele strength. This could be more relevant for complex and large optimisation problems where efficiency over time becomes important. Also, it could decrease the chance that a GA converges on a sub-optimal solution space, as it maintains and is aware of a full history of allele strength or combinations of alleles. Alternatively to PRAS,

which considers the current population, an historic records of alleles could be used to guide the mutation direction. If an allele historically had a low chance of leading to a positive mutation, then we could become more confident that in fact it's a negative allele. We don't have to be limited to what's possible in nature.

- **Problem formulation**

    - Extend the genotype with genes that affect the fitness of several alleles, such as by adding a gene for material choice. Since these genes are likely to have more interdependencies with other alleles, adding more of such genes would provide additional challenges for the design of heuristics.

    - Researching methods to accommodate proper levels of hierarchy in design optimisation using genetic algorithm genotypes, as well as proper levels of granularity. Traditionally, key domains such as structural integrity, propulsion and electrical systems have been separated in design. Yet, optimising across these boundaries, rather than isolated, can provide additional efficiency gains. How to incorporate all these domains and engineering models in a single genotype such that we can maintain a reliable GA optimisation would be paramount to capture the full potential of this optimisation technique.

8. APPENDIX

A. MATHEMATICAL DEMONSTRATION INEQUALITY CRENELLATION

The following equations were first documented in (Uz, et al., 2009) and provided a first hypothesis that the fatigue life of a flat plate can be improved through thickness variations.

$$N_{ref} = \frac{2\Delta a}{C\Delta K_a^m} \quad (A2)$$

$$N_{mod} = \frac{\Delta a}{C(\Delta K_a - t\Delta K_a)^m} + \frac{\Delta a}{C(\Delta K_a + t\Delta K_a)^m} \quad (A2)$$

$$\frac{2a}{C\Delta K_a^m} < \frac{a}{C(\Delta K_a - t\Delta K_a)^m} + \frac{a}{C(\Delta K_a + t\Delta K_a)^m} \quad (A3)$$

$$\frac{2a}{C\Delta K_a^m} < \frac{a}{C\Delta K_a^m}\left(\frac{1}{(1-t)^m} + \frac{1}{(1+t)^m}\right) \quad (A4)$$

$$2 < \frac{1}{(1-t)^m} + \frac{1}{(1+t)^m} \quad (A5)$$

$$f'(t) = m\left(\frac{1}{(1-t)^{m+1}} + \frac{1}{(1+t)^{m+1}}\right) \quad (A6)$$

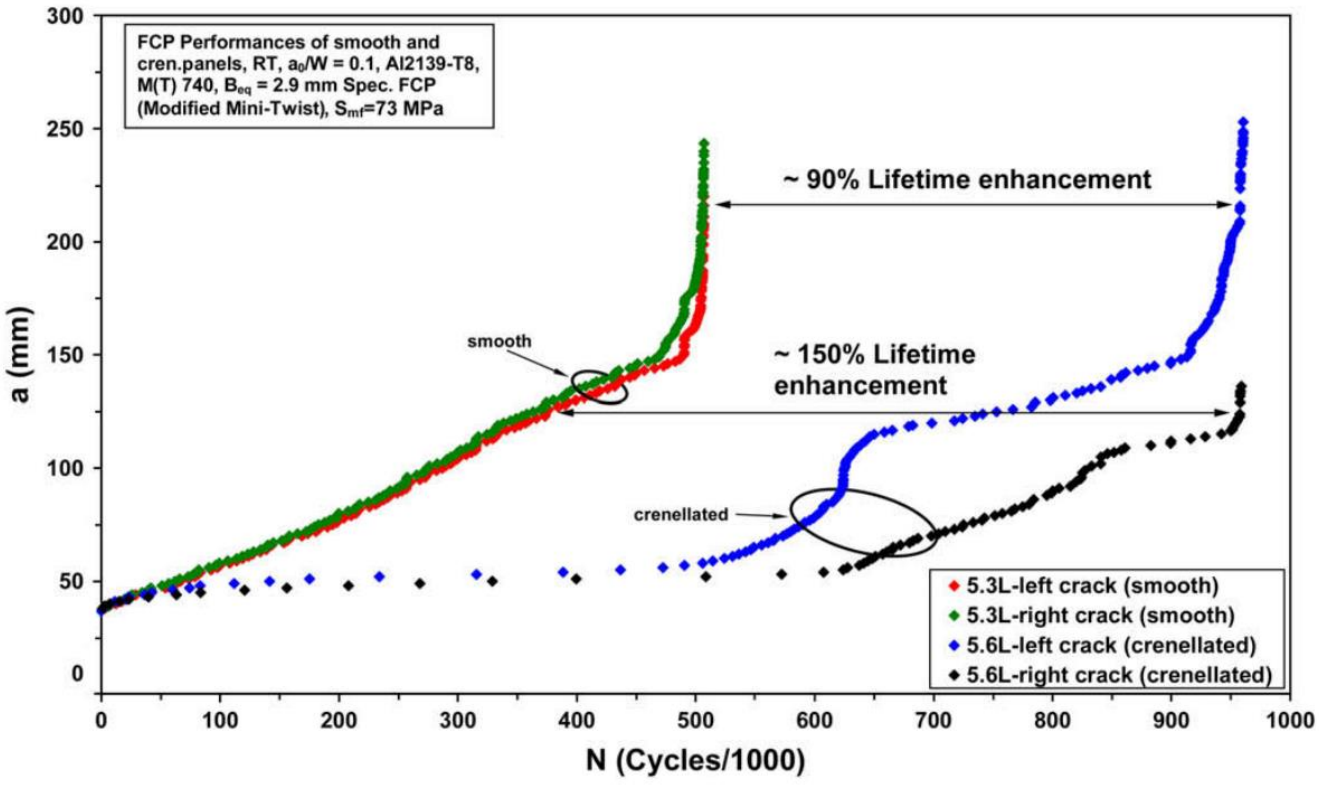


Figure 54 Comparison of a vs N curves of reference and crenellated panels under variable amplitude loading (Uz, et al., 2009)

B. ANALYTICAL FATIGUE MODEL

MODEL

First, consider that a load S_{max} is applied to the crenellated plate in Figure 8. As a result, a far-field stress σ_{skin} will be present in the plate.

$$\sigma_{skin} = \frac{S_{max}}{\int^{W} t_{s_i}(x)\, dx} \quad (1)$$

Assume that the stress state of a uniform thickness plate ahead of the crack tip can be expressed according to the Westergaard stress distribution, given as,

$$\sigma_{iso} = \frac{\sigma_{skin}}{\sqrt{1-\left(\frac{a}{x}\right)^2}} \; for\; a \le x \le W \quad (2)$$

The stress field σ_{iso} along the cracked, cross-x`x`sectional area must equal the total load applied.

$$S_{max} = \int_a^W \sigma_{iso} * t_{s_i}(x)\, dx \quad (3)$$

Then, substituting Equation 11 in Equation 12 gives the following expression for the effective stress σ_{eff} around the cross-sectional area in terms of the variable thickness function $t_{s_i}(x)$. We can take σ_{skin} outside of the integral as it is not dependent on crack length a and it represents the far field stress on the plate.

$$S_{max} = \sigma_{skin} \int_a^W \frac{t_{s_i}(x)}{\sqrt{1 - \left(\frac{a}{x}\right)^2}} dx \quad (4)$$

Rearranging for σ_{skin} for along the cross-sectional area gives us the following equation, which represents the far field stress in terms of crack length a and the cross-sectional area.

$$\sigma_{skin} = \sigma_{eff} = \frac{S_{max}}{\int_a^W \frac{t_{s_i}(x)}{\sqrt{1 - \left(\frac{a}{x}\right)^2}} dx} \quad (5)$$

The magnitude of the stress field in the cross-sectional area is dependent on the cross-sectional area A of the plate. The size of the cross-sectional area is dependent on the crack length a, since the crack will reduce the cross-sectional area when it grows.

According to the definition of the stress intensity factor, the value of K of a cracked plate can be written in terms of the far field stress σ_{eff}.

$$K = \sigma_{eff} * \sqrt{\pi a} \quad (6)$$

By substituting Equation 14 in Equation 15, we have can calculated the stress intensity factor around the crack tip based on a crenellated plate. The stress intensity factors can then be used to calculate the number of load cycles ΔN_i are needed to grow the crack with an increment of Δa_i , where the number of increments i ranges from $\left(1 , \frac{a_{max} - a_0}{\Delta a}\right)$, such that a crack growth rate can be determined,

$$\frac{da_i}{dN} = C * {K_i}^m \quad (7)$$

$$\Delta N_i = \frac{\Delta a_i}{\frac{da_i}{dN}} \quad (8)$$

Total fatigue life is then obtained by substituting Equation 16 in Equation 17 and sequentially integrating Equation 17 after the substitution.

$$N = \int_{a_0}^{a_{max}} \frac{da}{C * \Delta K^m} \tag{9}$$

Further substituting Equation 15 and subsequently Equation 14, we will arrive at the following expression of fatigue life,

$$N = \int_{a_0}^{a_{max}} \frac{\Delta a}{C * \left(\sigma_{eff}\sqrt{\pi a}\right)^m} = \int_{a_0}^{a_{max}} \frac{\Delta a}{C * \left(\frac{S_{max}\sqrt{\pi a}}{2 * \int_a^W \frac{t_{s_i}(x)}{\sqrt{1-\left(\frac{a}{x}\right)^2}} dx} \right)^m} \tag{10}$$

This rather extensive equation for the fatigue life N of a cracked, crenellated plate becomes difficult the interpret. For this reason we attempt to simplify it as much as we can.

Using the knowledge from the literature review on crack growth behaviour in crenellated panels, we aim to derive an analytical model suitable for predicting crack growth rates in crenellated panels.

$$N_{life} = \sum_{a_0}^{a_{max}} dN \tag{11}$$

By changing the material placement through manipulation of the thickness function $t_{s_i}(x)$, different values for the fatigue life can be achieved. The fatigue life is calculated based on a analytic model which calculates the number of fatigue cycles N are necessary to let the crack grow with an increment of Δa. The fatigue life increments ΔN can be summed, such that we arrive at the total fatigue life, described by Equation 20.

EFFECTIVE STRESS IN FRONT OF THE CRACK

Equation 14 shows an analytical solution for the effective stress for a given, load, crack length, thickness function and position x along the width of a panel.

Referencing to the denominator of equation 14, the evaluation of the crenellation area in front of the crack for each crack increment can be calculated using linear algebra. The integral can be considered a function $f(x)$, such that integration leads to,

$$f(x) = \frac{1}{\sqrt{1-\left(\frac{a}{x}\right)^2}} \;;\; F(x) = x\sqrt{1-\frac{a^2}{x^2}} + C$$

such that for one container n_i in front of the crack at a certain distance x_{n_i} from the centre of the crack the integral expression can be evaluated as,

$$\left[\int_w \frac{1}{\sqrt{1-\left(\frac{a}{x}\right)^2}}\,dx\right]_{n_i} = \left[x\sqrt{1-\frac{a^2}{x^2}} + C\right]_{x=x_{n_i}+\frac{1}{2}\Delta a} - \left[x\sqrt{1-\frac{a^2}{x^2}} + C\right]_{x=x_{n_i}-\frac{1}{2}\Delta a}$$

The total number of crack growth steps to be evaluated is equivalent to the maximum crack length divided by the step change in crack length, such that,

$$a_{tot} = \frac{a_{max}}{\Delta a}$$

Where the first crack length is equal to the initial crack length,

$$a_k = a_0$$

Build-up of a matrix A with the ranges from the initial crack length to the total crack set out against the width of the panel can be determined as,

$$\boldsymbol{A} = \begin{bmatrix} a_k \\ a_{k+1} \\ \vdots \\ a_{tot} \end{bmatrix} \begin{bmatrix} x_{n_i} & x_{n_{i+1}} & x_{n_{tot}} \end{bmatrix} \qquad 11$$

The indexes of matrix A determine the inputs of a and x for the evaluation of the integral in equation 7. The results of these integrations must be multiplied by the respective thickness of each container at an equal point and summated, such that,

$$At = B \qquad 12$$

Where t is a vector of thickness levels at each container (i.e. the crenellation pattern) and B a vector with the summation of the area in front of the crack for a certain crack length and crenellation pattern.

Equation 12 can be expanded for clarity purposes as follows,

$$\begin{bmatrix} \int_w \frac{1}{\sqrt{1-\left(\frac{a_k}{x_i}\right)^2}}dx & \int_w & \int_w \\ \int_w & \ddots & \ddots \\ \int_w & \ddots & \ddots \end{bmatrix} \begin{bmatrix} t_{n_i} \\ t_{n_{i+1}} \\ \vdots \\ t_{n_{total}} \end{bmatrix} = \begin{bmatrix} \sum_{n_a}^{n_{total}} t_{n_i} \int_w \frac{1}{\sqrt{1-\left(\frac{a_k}{x_i}\right)^2}}dx \\ \sum \\ \sum \end{bmatrix} \quad 13$$

It can be observed in equation 7 that when $a > x$, the square root will become a complex number as the term inside the square root will be negative. These values will be considered as 0, as their area is assumed to no longer contribute to the resistance of crack growth. This means that as soon as the crack length is equal to the position x_i of the container, the term will evaluate to 0. With a relative small container width Δx compared to the total panel width W, this should be a fair approximate to start with.

After these calculations, the effective stress can be evaluated for each crack length and crenellation pattern.

C. CALCULATION OF THE PRASF

The below schematic provides a visual example of how a mutation candidate list (CL) Glover et al, 2016 of two arbitrary alleles can be constructed based on RASF. Important to note from Figure X is that at no point do any of the GAs have knowledge about the optimal end-state of an allele as the algorithm only uses the definition of PRAS.

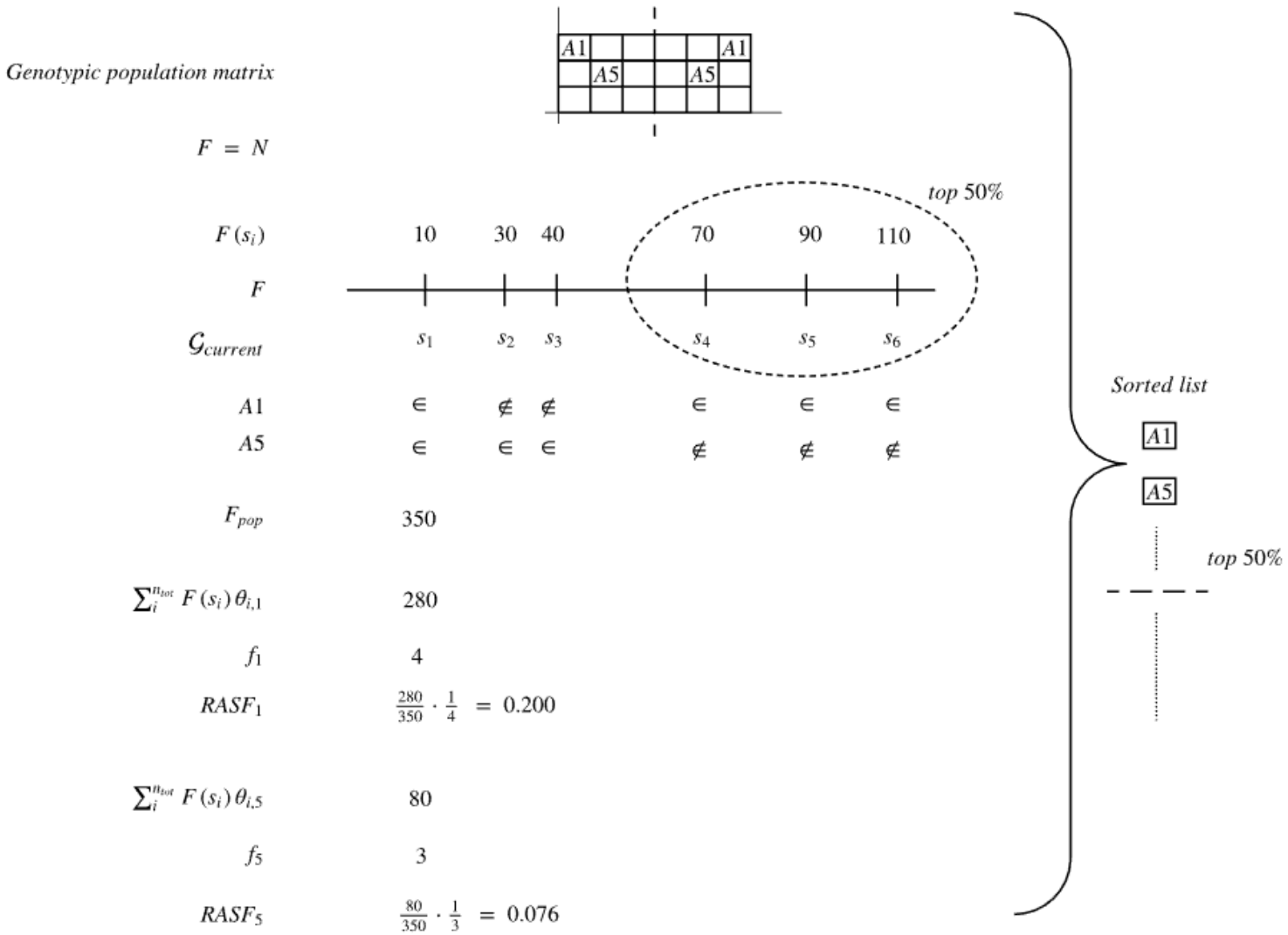


Table 14 Equivalent terms in mathematical and genetic algorithm optimisation

Mathematical	Genetic algorithm
Objective function	Fitness function
Value	Fitness
Solution	Individual
Set of solutions	Population
Decision variable	Gene
Feature	Gene
Decision variable value	Allele
Array of decision variables	Genotype
Array of decision variables	Chromosome
Engineering design	Phenotype
Initial set of solutions	Initial population
Old solutions	Parents
New solutions	Children / Offspring
Search heuristic	Crossover
Search heuristic	Mutation
Iteration	Generation

BIBLIOGRAPHY

Back, T., 1994. Selective pressure in evolutionary algorithms: a characterization of selection mechanisms. *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence,* Issue 1, pp. 57-62.

Back, T., Fogel, D. & Michalewicz, Z., 2000. *Evolutionary Computation 1: Basic Algorithms and Operators.* s.l.:s.n.

Bäck, T. & Hoffmeister, F., 1991. Extended Selection Mechanisms in Genetic Algorithms. *Proceedings of the 4th International Conference on Genetic Algorithms,* pp. 92-99.

Caraballo, S. C., Rodríguez, J. L. O. & Ruiz, J. A. L., 2017. Optimization of a butterfly valve disc using 3D topology and genetic algorithms.

Cheng, F. Y. & Li, D., 1998. Genetic Algorithm Development for Multiobjective Optimization of Structures. *AIAA Journal,* 36(6), pp. 1105-1112.

Corne, D. W. & Knowles, J. D., 2007. Techniques for highly multiobjective optimisation. *Proceedings of the 9th annual conference on Genetic and evolutionary computation - GECCO '07,* p. 773.

Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation,* 6(2), pp. 182-197.

Doerr, B., Happ, E. & Klein, C., 2012. Crossover can provably be useful in evolutionary computation. *Theoretical Computer Science,* Volume 425, pp. 17-33.

ElMaraghy, W., ElMaraghy, H., Tomiyama, T. & Monostori, L., 2012. Complexity in engineering design and manufacturing. *CIRP Annals,* 61(2), pp. 793-814.

Fischer, S. & Wegener, I., 2005. The one-dimensional Ising model: Mutation versus recombination. , , (), p. .

Fogel, D. B. & Atmar, J., n.d. Comparing genetic operators with gaussian mutations in simulated evolutionary processes using linear systems. p. .

Fonseca, C. M. & Fleming, P. J., 1993. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. *Proceedings of the 5th {I}nternational {C}onference on {G}enetic {A}lgorithms,* Issue July, pp. 416-423.

Fonseca, C. M. & Fleming, P. J., 1995. An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation,* 3(1), pp. 1-16.

Forrest, S. & Mitchell, M., 1993. Relative Building-Block Fitness and the Building-Block Hypothesis. *Foundations of Genetic Algorithms,* 1 1, Volume 2, pp. 109-126.

Friedrich, T., Hebbinghaus, N. & Neumann, F., 2007. Rigorous analyses of simple diversity mechanisms. , , (), p. .

Friedrich, T., Oliveto, P. S., Sudholt, D. & Witt, C., 2009. Analysis of diversity-preserving mechanisms for global exploration. *Evolutionary Computation,* 17(4), pp. 455-476.

Glover, F., 2016. Multi-wave algorithms for metaheuristic optimization. *Journal of Heuristics,* 3 6, 22(3), pp. 331-358.

Goldberg, D. E. (. E. & E., D., 1989. *Genetic algorithms in search, optimization, and machine learning.* s.l.:Addison-Wesley Longman Publishing Co., Inc..

Goldberg, D. E. & Sastry, K., 2001. A Practical Schema Theorem for Genetic Algorithm Design and Tuning. *Genetic and Evolutionary Computation Conference ({GECCO 2001}),* pp. 328-335.

Goldberg, D. & Sastry, K., n.d. Analysis of Mixing in Genetic Algorithms: A Survey. p. .

Holland, J. H., 1992. *Genetic Algorithms.* s.l.:Scientific American, a division of Nature America, Inc..

Jansen, T. & Wegener, I., 2005. Real royal road functions - Where crossover provably is essential. *Discrete Applied Mathematics,* 149(1-3), pp. 111-125.

Ling, K., 2003. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. , , (), p. .

Lu, J., 2017. Optimized Crenellation Designs for Fatigue Life Improvement of Metallic Airframe Structures. *PhD dissertation.*

Lu, J., Kashaev, N. & Huber, N., 2016. Optimization of crenellation patterns for fatigue crack retardation via genetic algorithm and the reduction in computational cost. *Engineering Failure Analysis,* Volume 63, pp. 21-30.

Marler, R. & Arora, J., 2004. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization,* 1 4, 26(6), pp. 369-395.

Mitchell, M., Forrest, S. & Holland, J. H., 1991. The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance. *Proceedings of the First European Conference on Artificial Life,* Issue 1, pp. 245-254.

Mitchell, M., Houand, J. H. & Forrest, S., n.d. When Will a Genetic Algorithm Outperform Hill Climbing?.

Moscato, P. & Cotta, C., n.d. A Gentle Introduction to Memetic Algorithms. *Handbook of Metaheuristics,* pp. 105-144.

Quinn, D., Murphy, A. & Cervi, L., 2011. Fatigue performance of aircraft panels with novel skin buckling containment features. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering,* 225(7), pp. 791-806.

Rans, C., Rodi, R. & Alderliesten, R., 2015. Analytical Prediction of Mode I Stress Intensity Factors for Cracked Panels Containing Bonded Stiffeners.

Rothlauf, F., 2006. *Representations for Genetic and Evolutionary Algorithms.* s.l.:s.n.

Roy, R., Hinduja, S. & Teti, R., 2008. Recent advances in engineering design optimisation: Challenges and future trends. *CIRP Annals - Manufacturing Technology,* 57(2), pp. 697-715.

Schijve, J., 2009. *Fatigue of Structures and Materials.* Delft: Springer Netherlands.

Senaratna, N. I., 2005. Genetic Algorithms: The Crossover-Mutation Debate. p. 22.

Shyalika, C., n.d. *An Insight to Genetic Algorithms - Part I.* [Online]
Available at: https://medium.com/datadriveninvestor/an-insight-to-genetic-algorithms-part-i-a7f5a5d6d214

Soni, N. & Kumar, T., n.d. Study of Various Mutation Operators in Genetic Algorithms. pp. 2-4.

Soremekun, G., Gürdal, Z., Haftka, R. & Watson, L., 2001. Composite laminate design optimization by genetic algorithm with generalized elitist selection. *Computers & Structures,* 1 1, 79(2), pp. 131-143.

Sorensen, K. & 2006. Variable neighbourhood search in commercial VRP packages: evolving towards self-adaptive methods. *,* 4, 16(), p. .

Sörensen, K., 2015. Metaheuristics-the metaphor exposed. *International Transactions in Operational Research,* 1, 22(1), pp. 3-18.

Sorensen, K., Sevaux, M. & Apr, A. I., 2017. A History of Metaheuristics. pp. 1-27.

Storch, T. & Wegener, I., 2004. Real royal road functions for constant population size. *Theoretical Computer Science,* 320(1), pp. 123-134.

Sudholt, D., 2018. The Benefits of Population Diversity in Evolutionary Algorithms: A Survey of Rigorous Runtime Analyses. 30 1.

Uz, M.-V.et al., 2009. Improvement of damage tolerance of laser beam welded stiffened panels for airframes via local engineering. *International Journal of Fatigue,* 5, 31(5), pp. 916-926.

Weise, T., Zapf, M., Chiong, R. & Nebro, A. J., 2009. *Why is optimization difficult?.* s.l.:s.n.

Whitley, D., Gordon, V. S. & Mathias, K., 1994. Lamarckian Evolution , The Baldwin Effect and Function Optimization.

Wolpert, D. H. & Macready, W. G., 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation,* 1(1), pp. 67-82.

Yang, S. & Zhao, Y. F., 2015. Additive manufacturing-enabled design theory and methodology: a critical review. *The International Journal of Advanced Manufacturing Technology,* 24 9, 80(1-4), pp. 327-342.