Biased Dyadic Crossover for Variable-Length Multi-Objective Optimal Control Problems

***Abstract*—This paper presents an enabling technique for social cooperation suitable for variable-length multi-objective direct optimal control problems. Using this approach, individualistic mesh-refinement may be performed across a population of discretised optimal control solutions within a real-coded evolutionary algorithm. Structural homology between individual solutions is inferred via the exploitation of non-uniform dyadic grid structures. Social actions, including genetic crossover, are enabled by identifying nodal intersections between parent vectors in normalised time. Several alternative crossover techniques are discussed, where effectiveness is evaluated based on the likelihood of producing dominating solutions with respect to the current archive. Each technique is demonstrated and compared using a simple numerical test case representing the controlled descent of a Lunar-landing vehicle. Of the examined methods, it is found that a hybrid one/two-point crossover, biased towards higher levels of grid resolution consistently outperforms those based on more traditional, unbiased crossover.**

***Index Terms*—Multi-Objective Optimal Control, Mesh Refinement, Evolutionary Algorithms, Variable-Length Chromosomes**

I. Introduction

A common issue throughout the field of evolutionary algorithms is that the restriction of flexibility in genotype length/structure required by standard combinatorial operators naturally precludes their use in more complex design scenarios [1]. Indeed, a traditional fixed-length representation in computational evolution may altogether sacrifice certain forms of genotypic variation otherwise common in natural genomes, thus potentially compromising the dynamics of the evolutionary process as a whole [2]. Examples of such are the duplication or removal of certain genes/sequences (a crucial process for the emergence of new genes in natural evolution [3], and intimately linked to genetic properties such as robustness, evolvability, and functional specialisation [4]), or the structural relocation of genes across the genotype [2].

From the computational perspective, variable-length representations of the problem genome [5] allow evolutionary modelling to mimic these phenomena to a higher degree. In consequence, this facilitates a more natural exploration of open-ended design problems where solution complexity and/or dimensionality are in themselves adaptive unknowns [6].

This may be particularly relevant when considering problems where solution fidelity/feasibility is inherently linked to the size of the associated genotype. For example, path optimisation problems where the number of waypoints is not known *a priori* [7], systems design problems where the number of discrete components is a optimisable parameter [8], or direct optimal control problems requiring adaptive mesh refinement to capture discontinuous/non-smooth behaviour in the system dynamics [9]–[11].

This paper expressly deals with this latter application, within the context of multi-objective optimisation. The following explores techniques for genetic crossover suitable for variable-length multi-objective direct optimal control problems, such that individualistic mesh-refinement may be performed concurrently within a real-coded evolutionary process.

For this class of problem, we introduce a method where structural homology is inferred via the exploitation of a non-uniform dyadic grid structure imposed across discretised dynamic profiles. This method is integrated into the Multi-Agent Collaborative Search algorithm [12], where a population of independent solutions may be propagated via both individualistic and social actions to converge on some optimal representation of the multi-objective trade-off surface. Considering social actions, crossover points and/or segments are identified via nodal intersections between parent solutions in normalised time. Crossover effectiveness is measured based on the likelihood of producing dominating solutions with respect to the current archive. Several distinct crossover methods are considered, each primarily differentiated by the number of crossover points and their selection strategy. For comparison, each is applied to a Multi-Objective Optimal Control Problem representing the descent of a Lunar-landing vehicle.

II. Background

Homology is defined in the field of evolutionary biology as "possessing a common evolutionary origin" [13]. This is as opposed to merely similarity of sequence, function or location. Considering linear genomes of constant vector length, genetic crossover is performed by exchanging homologous sequences between two parental genomes. The most prevalent methods are typically differentiated by the number n of crossover point-pairs selected to define such sequences (e.g. $n = 1, 2, \ldots, k$) [2]. Crossover points are often selected assuming unambiguous homology across successive generations. That is, each element or sequence remains fixed in position within a particular genome. Clearly, this assumption cannot hold for variable-length structures. As in nature, variable-length crossover must be guided to exchange sensible or equivalent sequences regardless of the respective lengths of the parent genomes. Practically, this may be achieved (for example) by embedding metadata within the genome [14], or by inferring homology via sequence similarity or some other relevant heuristic [15].

An example of the latter is the Species Adaptation Genetic Algorithm (SAGA) [1]. SAGA utilises a similarity metric, the

This is a peer-reviewed, accepted author manuscript of the following article: Parsonage, B., & Maddock, C. (2024). Biased dyadic crossover for variable-length multi-objective optimal control problems. In 2024 IEEE Congress on Evolutionary Computation (CEC) IEEE. https://doi.org/10.1109/CEC60901.2024.10611888

Longest Common Subsequence (LCSS), designed to favour the exchange of similar segments during recombination. Similarly to Goldberg's Messy-GA (mGA) [16], an initial crossover point is randomly selected from the first parent genome. However, where mGA randomly samples a corresponding point in the second parent genome, SAGA sequentially tests each possible matching point in the second parent. The pairing with the maximum sum of LCSS (on both the left and right split segments of each genome) is selected. Where there are multiple candidates with identical scoring, one is selected at random. A similar approach is adopted in the Virtual Virus algorithm [17], where crossover probability is regulated by the degree of local similarity between parent genomes, given by the number of matches within a specified fixed size window. However, the SAGA crossover approach explicitly attempts to preserve a complete genetic sequence using the parent genomes as a guide, and thus remains superior in this respect.

These methods fundamentally assume the genome itself is an inflexible, rigid array of data, allowing unbiased selection of the initial crossover point. As such, this selected point may lie within any unique structure or subsequence that has a particularly strong effect on the fitness of the genome as a whole, thus is less likely to have a complimentary (or even appropriate) pair in the second genome. This can result in a largely disruptive crossover, potentially compromising the exchange and/or preservation of data. In the work [18], a softer approach is demonstrated, more in line with biological crossover. The Synapsing Variable-Length Crossover (SVLC) algorithm selects the initial crossover point biased by the physical structure of *both* parent genomes. Parents are *synapsed* together at points of similarity (again determined using the LCSS approach, with longer sequences given priority). Crossover is then performed only within synapsed regions, producing offspring that inherit the entirety of the common parental subsequences. This potential for smoother, non-destructive crossover is particularly appealing given the findings of [1], who used the concepts of *epistasis* and *fitness landscapes* to show that propagation through a search space of indefinite dimensionality may only be feasible through a *gradual* increase in genotype length.

It may thus be suggested that the inference of homology via the genetic structure of parental genomes is more likely to facilitate meaningful, non-destructive crossover. In this manner, homologous segments may be both adaptively selected and compared throughout an open-ended optimisation process. It is via this principle that the following methods are proposed and applied within the context of evolutionary multi-objective optimal control, where the structure of unique solutions is inherently linked to the NLP transcription approach.

III. Multi-Objective Optimal Control with Variable-Length Structures

Multidisciplinary Design Optimisation (MDO) problems can often be formulated as Multi-Objective Optimal Control problems (MOCP). This is useful for systems where overall performance is comparably dependant on the time-dependant (dynamic) control law by which it must operate as it is on time-independent (static) system design parameters [19].

Solving MOCPs can become prohibitively expensive as complexity increases. Most practical examples include some number of state constraints, control constraints, and equality/inequality boundary and/or path constraints. As a consequence, solutions may exhibit rapid changes or discontinuities in state/control profiles and/or their respective higher order derivatives [11]. Considering *direct* transcription methods[1], solution accuracy can therefore be highly dependant on the grid resolution of the discretisation nodes at which the dynamic variables are evaluated. If the constraints are to be satisfied to within a strict tolerance, a high-resolution (dense) grid is typically required. However, this may induce an excessive computational cost given the resulting size of the NLP problem and the associated augmentation of search space dimensionality. Therefore, dense grids may remain impractical for most applications, especially if the NLP problem is not sparse [9], [10]. Furthermore, as grid density increases, the chosen solver may become ill-conditioned and exhibit poorer convergence characteristics, ultimately failing to converge at all [11].

Reducing the cost of MOCPs can be approached via a progressive *non-uniform* grid refinement. This entails the adaptive addition, subtraction, or redistribution of discretisation nodes to concentrate the grid around non-smooth regions (or, particular areas that dominate the overall solution accuracy), while smooth regions remain relatively sparse. Such approaches has been widely applied to single-objective problems [9]–[11], [21]. However, if we are to consider MOCPs, individual solutions (particularly those orthogonal to the objectives) may exhibit non-linear/non-smooth behaviour at different locations within the time domain. This implies that to efficiently represent the associated trade-off surface, each solution must be separately discretised upon it's own unique, non-uniform grid. Within the context of evolutionary computation, this then by definition prohibits the assumption of direct one-to-one correspondence between parameter loci across alternative solutions, precluding the use of traditional crossover techniques.

A. Mesh Refinement

In this study, a continuous MOCP is transcribed into a finite-dimensional NLP problem using Direct Finite Elements in Time (DFET) [22], where system state and control functions are approximated using Bernstein polynomials[2]. The collection of discretisation nodes τ that represent the interface between each finite element D_ϵ of a solution vector $\mathbf{x}_i$ over a normalised time interval may be considered a uniform dyadic mesh $\mathcal{V}_{j,N}$ conforming to the general form:

$$\mathcal{V}_{j,N} = \left\{ \tau_{j,k} \in [0,1] : \tau_{j,k} = k/(2^j N), \quad 0 \leq k \leq 2^j N \right\}, \quad 0 \leq j \leq J_{max} \tag{1}$$

[1]The reader is referred to [20] for a comprehensive overview of indirect vs direct transcription methods for optimal control.

[2]The reader is referred to [23] for a detailed description of this approach, and the various practical advantages of Bernstein polynomials that support their use in this application.

where $\tau_{j,k}$ are the time coordinates of the interface nodes, with corresponding spatial locations k. The positive integers j and J_{max} represent the current and maximum allowable resolution level and N is the number of elements present in the initial uniform mesh. A dyadic grid is obtained by successively subdividing a uniform grid such that $\mathcal{W}_{j,N}$ denotes the set of grid points belonging to $\mathcal{V}_{j+1,N}$ but not $\mathcal{V}_{j,N}$; that is,

$$\begin{aligned}\mathcal{W}_{j,N} = \{\hat{\tau}_{j,k} \in [0,1] : \hat{\tau}_{j,k} = (2k+1)/(2^{j+1}N),\\ 0 \leq k \leq 2^j N - 1\}, \quad 0 \leq j \leq J_{max} - 1\end{aligned} \tag{2}$$

Hence, $\tau_{j+1,k} \in \mathcal{V}_{j+1,N}$ if:

$$\tau_{j+1,k} = \begin{cases} \tau_{j,k/2}, & k \text{ is even} \\ \hat{\tau}_{j,(k-1)/2}, & \text{otherwise} \end{cases} \tag{3}$$

N can be any positive integer for the generalised dyadic grid, which is more convenient for optimisation. An example of a uniform dyadic grid with $N = 1$ and $J_{max} = 7$ is shown in Fig. 1. The subspaces $\mathcal{V}_{j,N}$ are nested;

$$\mathcal{V}_{0,N} \subset \mathcal{V}_{1,N} \subset \cdots \subset \mathcal{V}_{J_{max},N} \tag{4}$$

with

$$\lim_{J_{max} \to \infty} \mathcal{V}_{J_{max},N} = [0,1] \tag{5}$$

The sequence of subspaces $\mathcal{W}_{j,N}$ satisfies the property $\mathcal{W}_{j,N} \cap \mathcal{W}_{l,N} = \varnothing$ for all $j \neq l$. The sets $\mathcal{V}_{j,N}$ and $\mathcal{W}_{l,N}$ for $j \leq l$ are orthogonal to each other, as are the sets $\mathcal{W}_{j,N}$ and $\mathcal{W}_{l,N}$ for $j \neq l$ (see Fig. 1). If element subdivision is performed in a non-uniform manner, (1) can be modified to represent the resulting non-uniform grid:

$$\begin{aligned}\mathcal{G} = \{\tau_{j_i,k_i} \in [0,1] : 0 \leq k_i \leq 2^{j_i}\},\\ J_{min} \leq j_i \leq J_{max} \text{ for } i = 1, \ldots, R\\ \tau_{j_i,k_i} < \tau_{j_{i+1},k_{i+1}} \text{ for } i = 1, \ldots, R-1\end{aligned} \tag{6}$$

where $\mathcal{G} \subset \mathcal{V}_{J_{max},N}$ and R represents the number of resolution levels present. Of note, working with dyadic grids is equivalent to employing interpolating wavelets for analysis of the underlying function [10] and thus retains a major benefit of wavelet-based analyses: multi-resolution functional representations.

In this work, mesh quality is assessed by estimating the local error. That is, the difference between the computed solution and the solution of the differential equation which passes through the computed point. The reader is referred to [22] for a detailed description and supporting derivation of this approach. Using this method, any additional procedure or integration is not required, as enough information can be derived from the discontinuities across the boundaries of each element.

Mesh-refinement is implemented at a user-defined frequency. Additional interface nodes are inserted at the midpoints of all elements not satisfying a user-specified error tolerance ϵ_{tol}, thus preserving the dyadic structure.

B. Multi-Resolution MACS

This work employs the Multi-Agent Collaborative Search (MACS) algorithm, a dominance-based memetic solver for MOO problems [12]. MACS initialises a population of virtual agents at random locations in the search space. Each agent in turn explores it's local neighbourhood through a series of individualistic actions. The population as a whole then performs social actions to concurrently advance towards the Pareto-front. The MACS algorithm additionally utilises a bi-level optimisation routine, as presented in [24], to maintain local feasibility and balance the global/local search. The alternating individual and social actions comprise the globally searching outer-level of the algorithm. Each action in turn generates a candidate solution and submits it to the inner-level. The inner-level then triggers a gradient-based optimisation step to ensure the candidate is feasible with respect to differential, path, and boundary constraints. The new candidate is then passed back to the outer level for assessment. Candidates that demonstrate an improvement are saved to the current population. After every user-specified number of iterations (and as a final step before the algorithm terminates) a further gradient-based optimisation step is triggered to locally refine the population using a set of scalarised single-objective sub-problems. The algorithm continues alternating social and individualistic actions, with periodic local refinement, until a maximum number of iterations, or objective function calls, is met. The reader is referred to [23] for a complete description of this approach.

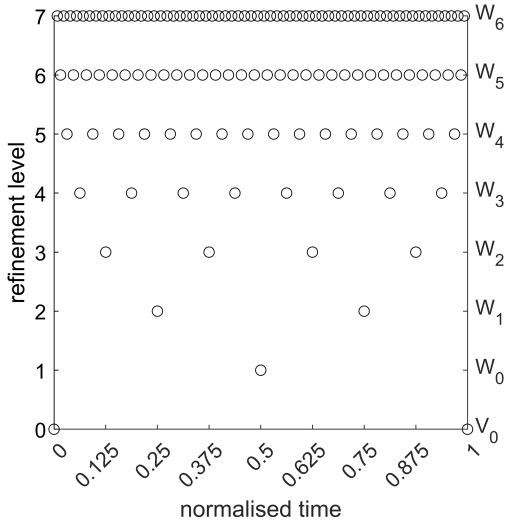


Fig. 1: A uniform dyadic grid where $N = 1$ and $J_{max} = 7$.

This work offers an extension to the MACS algorithm in which the proposed mesh-refinement procedure is periodically applied to the current population of P agents $\mathbf{x}_p, (1 \leq p \leq P)$. Each agent represents a solution to a MOCP initially discretised upon a uniform grid $\mathcal{V}_{j,N}^{(p)}$ using the DFET transcription process [22], [23]. By imposing the condition that any discretised element may only be split exactly in half, the population evolves into a set of unique, variable-length solutions defined upon *multi-resolution* (non-uniform) dyadic grids $\mathcal{G}^{(p)}$.

C. Variable-Length Crossover

The nodal intersection between two or more unique dyadic grids can be easily computed. Considering discretised MOCPs,

this allows common vector components to be quickly identified, thus facilitating direct comparison and/or recombination for the purpose of social action. It is important to note that given the initial uniform discretisation of each agent, the proposed dyadic refinement implies that any segment bounded by common intersection nodes of any two agents must by definition share common ancestry. With reference to our previous definition (Section II), *homology* in this context additionally infers that such segments represent the same portion of the solution in normalised time, regardless of the number of intermediate nodes enclosed within.

Consider two individual solution vectors of respective lengths L and K, $\mathbf{x} = [x_0, x_1, \ldots, x_i, \ldots, x_L]$ and $\mathbf{x}^* = [x_0^*, x_1^*, \ldots, x_j^*, \ldots, x_K^*]$ with corresponding multi-resolution grids $\mathcal{G}^{(\mathbf{x})}$ and $\mathcal{G}^{(\mathbf{x}^*)}$. Given that each agent within the population is initialised upon an identical uniform grid, i.e. $N^{(\mathbf{x})} = N^{(\mathbf{x}^*)}$, and that $z = 0.5$, the grid intersection between $\mathbf{x}$ and $\mathbf{x}^*$ will include at least one intermediate node (excluding the trivial case of common initial and final nodes) such that:

$$\mathcal{G}^{(\mathbf{x})} \cap \mathcal{G}^{(\mathbf{x}^*)} \neq \varnothing, \quad 2 \leq R^{(\mathbf{x},\mathbf{x}^*)} \leq 2^{J_{max}} \tag{7}$$

The simplest social action that can thus be considered is classical one-point crossover (1PCO). In this case, a single intersection node is selected at random, $x_i = x_j^*$, with all components following x_i and x_j^* exchanged between the parent solutions to create two new individuals[3].

One-point crossover has the feature that solution components nearer the vector extremes are much more likely to be separated than components nearer the middle [1]. Intuitively, two-point crossover (2PCO) may avoid this inherent bias by instead randomly selecting two intermediate nodes from the set of intersections between $\mathbf{x}$ and $\mathbf{x}^*$, exchanging the enclosed subsequence. In this case, grid intersections conform to:

$$\mathcal{G}^{(\mathbf{x})} \cap \mathcal{G}^{(\mathbf{x}^*)} \geq 2, \quad 2 \leq R^{(\mathbf{x},\mathbf{x}^*)} \leq 2^{J_{max}} \tag{8}$$

A known flaw within both methods is the potential for lack of offspring diversity should the crossover point(s) be chosen too close to the initial and/or final nodes [7]. In such cases, this is equivalent to saying the length of the exchanged segment is comparable to the length of the parent vectors themselves. An interesting consideration within the context of this study is that as the population is progressively evolved, the number of nodal intersections between similar solutions is likely to increase. This therefore allows a progressively finer level of crossover to occur between similar solutions, thus reducing the overall difference between agents comprising successive generations. This itself has implications for the convergence characteristics of the algorithm as a whole.

However, whilst the potential for smoother social cooperation is introduced, crossover points are still selected at random. This means that there is no guarantee that crossover at a higher level of resolution will actually occur. Indeed, [1] presents a modified 2PCO that chooses the second point such that the similarity between the exchanged segments is maximised. In the current work, a bias is introduced weighted towards the exchange of higher resolution segments. As the number of intersection points between solutions increases, information exchange will occur at as high a resolution level as is available. This aims to minimise the potential disruptive effects caused by blind exchange of larger subsequences (a not uncommon feature of unbiased crossover) and thus promotes a more gradual convergence in objective space. Furthermore, if the selection of intersection points is allowed to include either the initial or final nodes, the biased process (referred to now as *dyadic crossover*, or DCO), can effectively switch freely between 1PCO and 2PCO classifications, guided by the resolution of the parent individuals.

IV. Computational Experiments

This section presents a computational test case intended to explore the effectiveness of the proposed approach. An example MOCP is transcribed using DFET and solved with the modified MACS algorithm (incorporating the bi-level optimisation, as described in Section III-B), where the original social operations are replaced with each crossover method described in Section III-C (1PCO, 2PCO and DCO) in turn. Two additional methods are included for reference. The first exchanges only the static parameters between the two selected parent solution vectors. In general, the static portion (e.g. design variables) of an optimal control solution vector is greatly outnumbered by the dynamic parameters (representing the system states and control law). As such, the lengths of the exchanged segments are comparable to those of the parent vectors themselves, thus this method is referred to here as 'near-uniform' crossover (NUCO). The second demonstrates an extreme case in which parent agents are simply swapped prior to the inner-level optimisation step, thus named 'no crossover' (NCO). In effect, this action represents a local optimisation step along a search direction different to that originally assigned to that particular agent[4]. The complete set thus represents a clear hierarchy of complexity, whilst including the most prominent fundamental methods from the literature (i.e. 1PCO and 2PCO). This allows the individual effects of each level to be more explicitly observed.

Given the prevalence of additional heuristics within MACS, it can be difficult to disseminate the global effects of any particular operation. For example, deficiencies in one may be compensated for by another, for a particular problem. Furthermore, the deliberate isolation of heuristics for testing purposes may not give an accurate representation of their behaviour when employed in collaboration. In an attempt to provide as isolated a performance metric as practical, the number of dominating solutions produced by the crossover action (including the inner-level gradient-based optimisation step of the bi-level approach) with respect to the current population is recorded as a measure of effectiveness. The

[3]This is functionally equivalent to the *cut* and *splice* operations introduced in Goldberg's Messy-GA [16].

[4]The reader is referred to [24] for a description of how the search direction is defined for each agent within the context of the MACS algorithm

likelihood of each approach to produce meaningful, globally exploratory offspring may thus be tracked in a relatively unbiased manner. Additionally, the quality and spread of the final Pareto-front is assessed against a reference Pareto-optimal set of solutions via the Inverted Generational Distance (IGD) and Averaged Hausdorff Distance[5] (AHD) metrics [25].

Due to the inherent stochastic nature of the employed algorithms, 100 independent analyses were performed to track both the mean and standard deviation of relevant metrics. Additionally, to objectively indicate statistical significance, a one-way Analysis of Variance (ANOVA) was performed for each metric throughout the computation, where the significance level required to reject the null-hypothesis is taken as $\alpha = 0.05$. To subsequently assess the extent of the variation between individual pairs of crossover methods, a multiple comparison test was performed using the ANOVA output.

A. Settings

All computations were performed using the ARCHIE-WeSt High Performance Computer. Individual runs utilised a single Intel Xeon Gold 6138 20 core 2.0GHz CPU, where 40 nodes comprise a standard compute node (on Lenovo SD30 servers) of 192GB RAM per node.

B. Formulation

The following MOCP represents the controlled descent of a Lunar-landing vehicle, based on the formulation originally presented in [26]. The problem is to determine the control law $\mathbf{u}$ that minimises the total impulse acting on the vehicle and the maximum allowable control force:

$$\min_{t_f,\mathbf{u}}[J_1, J_2]^T = \left[\int_{t_0}^{t_f} u \ \mathrm{d}t, u_{max}\right]^T \tag{9}$$

The system state dynamics are given by:

$$\dot{h} = v \tag{10}$$

$$\dot{v} = -g + u \tag{11}$$

where h [m] is altitude, v [m/s] is velocity, g [m/s^2] is the gravity force and u [m/s^2] is the upwards control force. Boundary conditions are given as:

$$\begin{aligned} h(0) = h_0 = 10, \quad v(0) = v_0 = 2 \\ h(t_f) = h_f = 0, \quad v(t_f) = v_f = 0 \end{aligned} \tag{12}$$

with a path constraint on the control force defined by:

$$0 \leq u \leq u_{max} \tag{13}$$

The gravity of the Moon is taken as $g = 1.5$ m/s^2, the final time t_f is free and the maximum control force is an optimisable static parameter $u_{max} \in [0, 3]$.

This problem was solved by initialising a population of 10 individual agents, where each is a single continuous time phase discretised upon a uniform mesh of resolution level $J = 0$ (i.e. a single finite element). States and controls are represented by element specific 2nd order Bernstein polynomials. The target mesh error imposed at element boundaries was set to $\epsilon = 10^{-5}$, with a refinement step performed every 10 iterations. Elements not meeting this value were split at a frequency coinciding with the gradient-based local searches (every 10 iterations). In this manner, local optimality may be guaranteed prior to each mesh refinement step. Local (inner-level) optimisation was completed using the MATLAB® fmincon solver with a SQP algorithm where function, constraint violation and step tolerances set to 1e-6, 1e-12 and 1e-9 respectively. The MACS algorithm was set to terminate upon reaching a maximum number of outer-loop iterations $n_{iter} = 200$.

Figure 2 shows an example representation of the Pareto front obtained using the proposed approach. Included are the corresponding state and control profiles, where circle markers represent element boundaries. This problem was chosen in part due to the inherent simplicity in formulation, whilst demonstrating several features of interest to the current study. For example, in Fig. 2 it can be seen that all solutions exhibit one clear discontinuity, corresponding to a *bang-bang* control structure and therefore suitable for analysis via an adaptive mesh approach. Furthermore, the location of this discontinuity in time changes as solutions move across one end of the Pareto-front to the other, a feature that should be captured by an effective variable-length crossover. This problem therefore represents a suitable platform to demonstrate and assess the *differences* between the included crossover approaches.

C. Results

Figure 3 shows the accumulation of dominating solutions generated by each of the described crossover approaches. Several interesting features can be observed. Firstly, 1PCO and 2PCO exhibit rather similar rates of change. The gradual plateauing of each may be attributed to the associated evolution of NLP size (shown in Fig. 4) which causes an increase in available grid point intersections and thus a decrease in the effectiveness of random point-pair selection. The effect of introducing the resolution-based bias can be clearly seen in the comparison between 2PCO and DCO, where the latter maintains an almost linear accumulation even as NLP size (and thus also the available grid intersections) begins converging. This may be due to the pronounced discontinuities seen in Fig. 2, in addition to the free (rather than fixed) final time t_f. The latter is important as the subsequent exchange of homologous segments identified in *normalised* time implies the exchange of *form* rather than simple time-equivalent numeric magnitudes. In this case then, the exchange of information via intersection-based crossover may have a higher potential of preserving/propagating important subsequences throughout the population. As expected, the NCO approach represents a performance baseline where no social manipulation is actually present. The generality of NCO may go to explain the linear accumulation (albeit at a reduced rate) of dominating solutions proportional to NLP size. Particularly interesting behaviour is seen from NUCO, initially accumulating non-dominated solu-

[5] As remarked by [25], the IGD metric is sensitive to the number of elements in the reference Pareto front and in the computed one, hence the inclusion of the AHD in the presented comparisons.

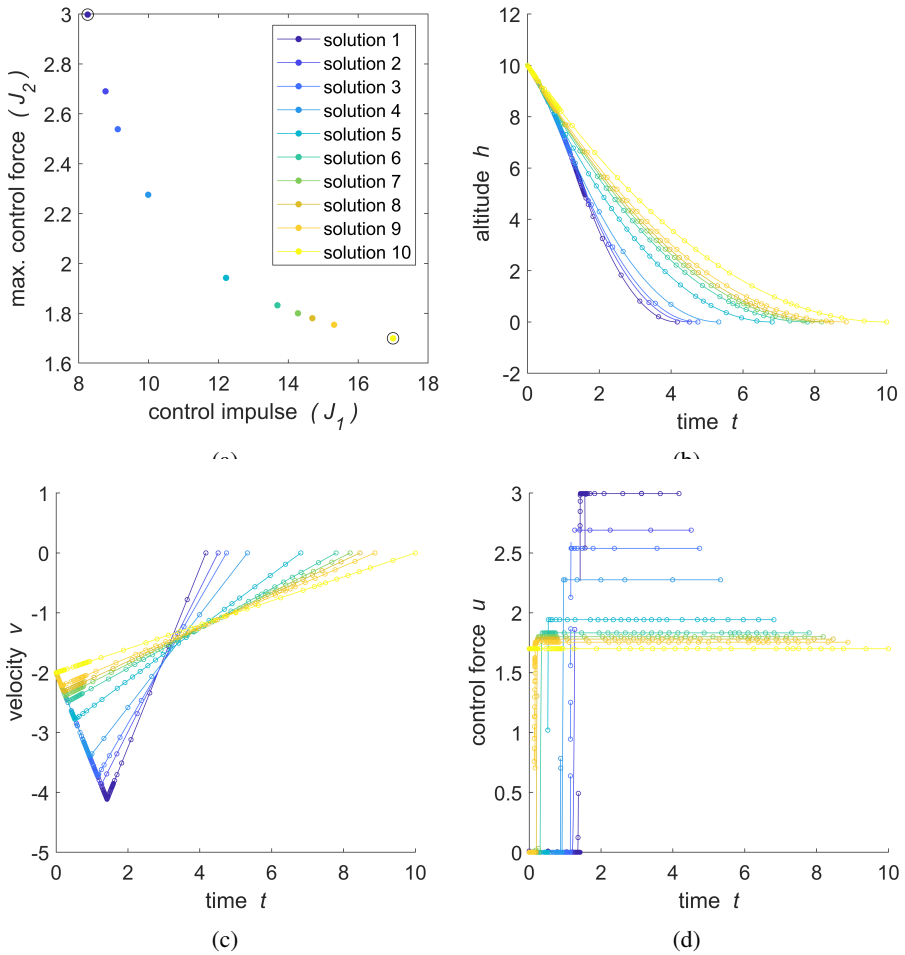


Fig. 2: Multi-objective solutions, including (a) Pareto front and corresponding (b-c) state and (d) control profiles.

tions at a higher rate than DCO, before exhibiting a plateauing similar to that of 1PCO and 2PCO. Given that NUCO is only exchanging static parameters (e.g. the free final time t_f), the initially larger accumulation rate may be attributed to the more exploratory nature of earlier iterations, where the choice in static parameters plays a more definitive role in the positioning of agents within objective space. As the distribution of static parameter values across the population stabilises, the rate of accumulation reduces towards the baseline level demonstrated by the NCO approach, interesting in the sense that the local search direction for NUCO *is* in the originally associated search direction.

The NLP size l_x and the number of dominating solutions (with respect to the current population) produced by the crossover operation n_{soc} at the final iteration ($n_{iter} = 200$) of each approach are shown in Table I, as well as the total recorded execution time t_f. Reported are the mean values across 100 separate analyses (standard deviation included in brackets). A clear hierarchy can be seen, where the dyadic approach seems to offer significant improvement relative to the others. Indeed, DCO exhibits execution times approaching the values of NCO, which itself could be taken as the theoretical minimum given that no recombination actually occurs.

Specifically considering the evolution of NLP size, two distinct groups are present (see Fig. 4). Interestingly, the first includes the single-point methods[6] and the second includes the multi-point methods. Whilst a lower average NLP size may suggest a generally more efficient distribution of grid nodes using DCO, the larger quantity and spread of outliers nonetheless suggests a remaining potential for excessive NLP 'bloat' (an acknowledged risk with variable-length representations in evolutionary algorithms [2]). Of note here is that in this work, NLP bloat is treated only indirectly through the user-specification of mesh-error tolerance. That is, an appropriate tolerance value prevents the algorithm from actively adding excessive new nodes. However, it is acknowledged that this does not prevent heavily refined segments from being relocated by the crossover mechanism to areas within the solution vector that do not require such levels of detail. The explicit identification and removal of unnecessary nodes is a complex problem and thus originally considered outwith the scope of this study, though remains an important subject of future work.

Perhaps a more surprising aspect of Fig. 4(b) is the range

[6]For the purpose of comparison, NCO can be considered a single-point method assuming the selected point is either of the extremal vector loci.

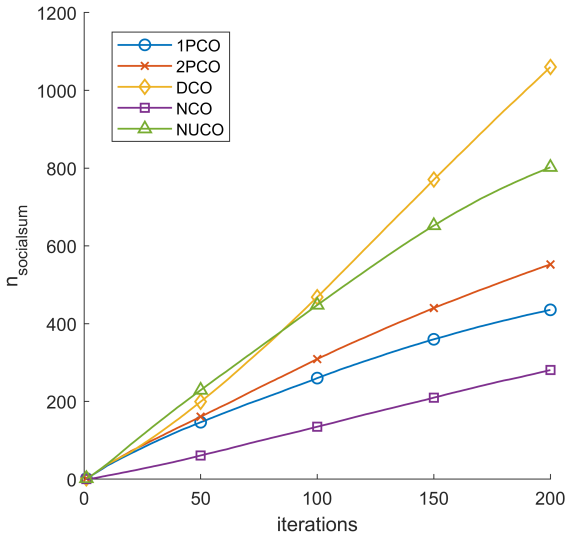


Fig. 3: Total number of dominating solutions accumulated by each crossover method.

TABLE I: Values of total execution time t_f, NLP size l_x and number of dominating solutions n_{soc} produced by each crossover operation at the final iteration $n_{iter} = 200$.

method	t_f [s]		l_x		n_{soc}	
1P	10496	(3.11e+3)	669	(196)	1.22	(1.03)
2P	8944	(2.67e+3)	584	(218)	2.19	(1.31)
D	7761	(2.55e+3)	582	(238)	5.82	(1.58)
N	7572	(2.90e+3)	673	(177)	1.34	(0.82)
NU	9754	(2.58e+3)	663	(177)	2.49	(1.30)

in final NLP size. Indeed, for each method the observed range is a significant proportion of the maximum observed value. As stated earlier, an important feature of the original mesh refinement problem is that individual solutions distributed across the Pareto-front may require unique discretisation grids to fully represent the corresponding state and/or control profiles (see Fig. 2). To quantitatively measure the extent of this requirement, we define the *grid intersect ratio* (GIR) as the fraction of intersection points between any two individuals relative to the total number of possible intersections. Figure 5(b) shows an example of the pairwise correlation in GIR recorded by the DCO method at the final iteration. The concentration of higher intersect values surrounding the diagonal of the agent pairing matrix supports the conjecture that similar solutions in objective space share a higher fraction of common nodes. Furthermore, the lowest observed value of GIR indicates that only around 30% of possible nodes are shared between the 'least similar' individuals of the final population, which may support the large range in final NLP size seen in Fig. 4(b).

To indicate how GIR varies between each method, Fig. 5(a) shows the respective distributions recorded at ($n_{iter} = 200$). One might expect a higher GIR to suggest a decrease in structural diversity across the population. That is, if two grids share a larger number of nodes, they are inherently less unique (a GIR of 1 indicates one is a complete subset of the other). However, a complicating factor is the associated NLP size. If a population exhibits a lower GIR but a larger average NLP size, this could suggest over-refinement. Conversely, a higher GIR coupled with a lower NLP size may suggest under-refinement, in that the population may not yet have achieved a genetic diversity complementary to the problem complexity. In part these results suggest the problem is relatively simple for the optimiser (albeit assuming some initial spreading in objective space), or at least is inexpensive enough to complete a sufficient number of iterations. Whilst as expected given the additional heuristics within MACS, this is nonetheless supported by the general agreement in IGD and AHD values between methods, see Table II.

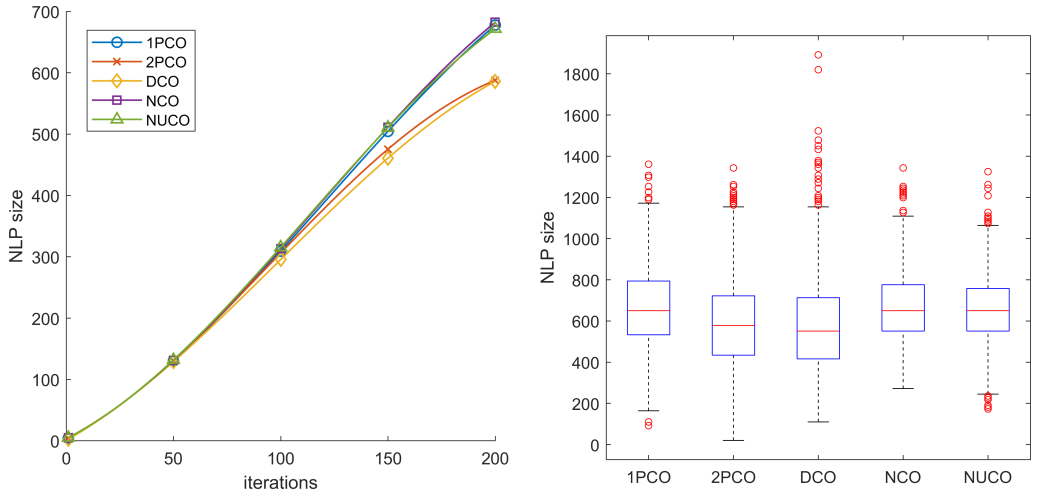
TABLE II: Mean values (standard deviation in brackets) of IGD and AHD for each crossover method.

Method	IGD		AHD	
1P	0.0927	(3.32e-2)	0.5290	(1.11e-1)
2P	0.0926	(2.90e-2)	0.5367	(1.06e-1)
D	0.0940	(2.81e-2)	0.5298	(9.40e-2)
N	0.0921	(2.50e-2)	0.5029	(1.08e-1)
NU	0.1016	(4.18e-2)	0.5505	(1.16e-1)

This agreement and it's relation to the complexity of the example problem can be further understood by considering the results of the statistical analyses. Table III shows the results of a multiple comparisons test (using Tukey's honestly significant difference criterion), where the first column indicates the specific pair of methods being directly compared through ANOVA, and the remainder displaying the p-values for each performance metric at $n_{iter} = 200$. Firstly, it can be seen that the numbers of non-dominated solutions n_{soc} produced by the DCO method is statistically significant relative to every other method. Indeed, it was found that only 40 intermediate iterations were required for this to be true. Secondly, the p-values reported for NLP size support the grouping seen in Fig. 4(a), in that the null-hypothesis can only be rejected between a single-point method and a multi-point method. A similar distinction is seen for GIR. Finally, and perhaps most importantly, the p-values of AHD and IGD suggest that there is no significant difference between the quality of final population obtained using any of the included crossover methods. When considered as a whole, these results ultimately suggest that the DCO method is exhibiting a meaningful improvement in the exploration of the design space, but not enough to affect the final convergence of the algorithm in the presence of the additional heuristics. This again highlights the need for more complicated test problems to assess whether the observed effectiveness of DCO (even if strictly internal in this case) is maintained.

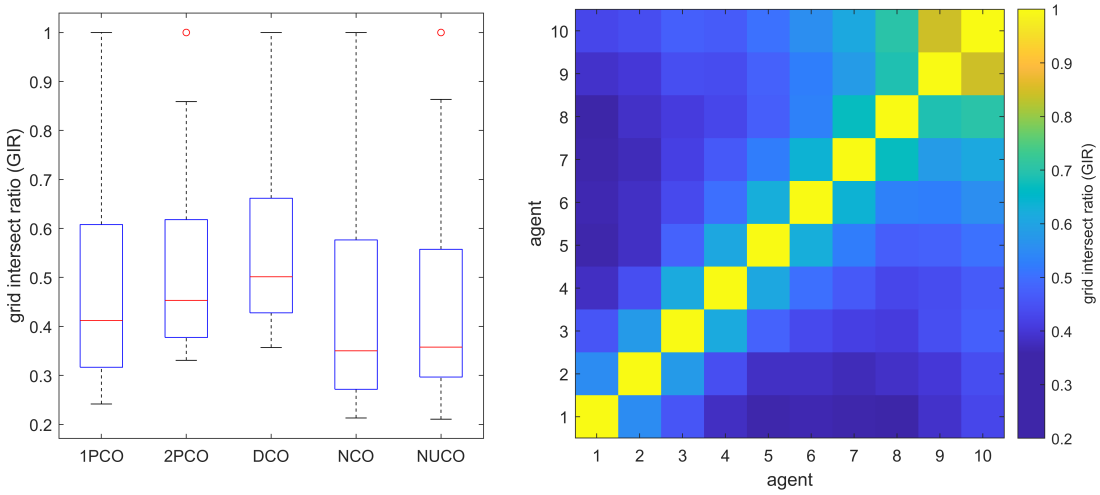
V. Conclusions

In general, it is found that the proposed approaches successfully promote offspring diversity throughout an open-ended evolutionary process. This is achieved by allowing a progressively finer level of social action to occur between similar solutions, thus reducing the overall genetic difference between



(a) (b)

Fig. 4: NLP size progression recorded during 100 separate runs using each crossover method, where (a) shows the fourier-fit trend of the averages across all iterations, and (b) shows the distributions at the final iteration.



(a) (b)

Fig. 5: Grid Intersect Ratio (GIR) recorded at the final iteration ($n_{iter} = 200$), where (a) shows the distribution of average values for each crossover method, and (b) shows the pairwise GIR across an example population solved using the DCO method.

TABLE III: ANOVA multiple-comparisons test.

Methods		p-value ($n_{iter} = 200$)				
		n_{soc}	NLP size	GIR	AHD	IGD
1P	2P	3.67e-6	5.50e-19	0.00	0.96	0.76
1P	D	0	1.48e-20	6.16e-9	0.51	0.37
1P	N	0.90	0.99	0.95	0.95	1.00
1P	NU	2.96e-8	0.96	0.78	0.71	0.98
2P	D	0	1.00	9.29e-2	0.89	0.97
2P	N	2.77e-4	1.53e-21	9.84e-5	1.00	0.80
2P	NU	0.92	5.60e-16	1.24e-5	0.97	0.97
D	N	0	2.59e-23	4.47e-11	0.92	0.41
D	NU	0	2.49e-17	1.73e-12	1.00	0.74
N	NU	4.84e-6	0.82	0.99	0.98	0.99

successive generations. A positive effect on the execution time of the algorithm is observed, as well as a reduction in the average NLP size across the population for the same solution quality. It is found that a bias towards exchanging higher resolution segments consistently produces more dominating solutions than traditional, unbiased crossover. However, the effects of these improvements on the final solution are found to be negligible, likely due to the simplicity of the example problem. Future work should include more complex scenarios such that crossover effectiveness becomes more important to the host algorithm. Excessive NLP growth remains a pressing issue outwith passive/indirect mitigation. Future study could include an adaptive approach to validate growth, detect over-refinement and ultimately remove unnecessary nodes without incurring additional expense or loss of crucial information.

A significant factor in these findings is the influence of additional heuristics within the host algorithm. For example, even if the crossover operation successfully produces a dominating solution with respect to the current population, inclusion into the population is not guaranteed without first demonstrating (via substitution with an existing archived individual) a collective improvement as per the archiving strategy implemented within MACS [27]. This further reliance on the *positioning* of candidate solutions within objective space, as opposed to merely the diversity or quality of the genetic information contained within, may significantly dampen the exchange/redistribution of genetic features across successive generations. This in turn affects the genetic material available for redistribution during the next iteration. Whilst partly analogous to a natural evolutionary process, this fundamentally makes it difficult to quantitatively isolate and evaluate the global effects of any particular operation, including the proposed crossover methods. It is of course true that this issue may not be fully due to the inherent complexity of the host algorithm. To maximise the efficiency of an evolutionary optimisation, high-fidelity is necessary to preserve homologous information. However, this feature of crossover is common, and can indeed be achieved even by simple algorithms [2]. We may conclude then that despite these misgivings, the results show that the inference of homology via the genetic structure of parental genomes is at least more likely in this case to allow and facilitate meaningful, non-destructive crossover through the redistribution of genetic information. The presented methodology may offer comparable savings in computational expense as would normally be associated with discipline-based multi-fidelity optimisation. Specific investigations into this aspect remains the subject of future work.

REFERENCES

[1] I. Harvey, "The SAGA Cross: The Mechanics of Recombination for Species with Variable-Length Genotypes," *Parallel Problem Solving from Nature*, vol. 2, pp. 269–278, 1992.

[2] A. Merlevede, H. Åhl, and C. Troein, "Homology and linkage in crossover for linear genomes of variable length," *PLoS ONE*, vol. 14, no. 1, pp. 1–16, 2019.

[3] M. Long, E. Betrán, K. Thornton, and W. Wang, "The origin of new genes: glimpses from the young and old," *Nature Reviews Genetics*, vol. 4, no. 11, pp. 865–875, 2003.

[4] R. Calabretta, S. Nolfi, D. Parisi, and G. P. Wagner, "Duplication of modules facilitates the evolution of functional specialization," *Artificial life*, vol. 6, no. 1, pp. 69–84, 2000.

[5] J. R. Koza, *Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems*. Stanford University, Department of Computer Science Stanford, CA, 1990, vol. 34.

[6] W. Banzhaf, G. Beslon, S. Christensen, J. A. Foster, F. Képès, V. Lefort, J. F. Miller, M. Radman, and J. J. Ramsden, "Guidelines: From artificial evolution to computational evolution: A research agenda," *Nature Reviews Genetics*, vol. 7, no. 9, pp. 729–735, 2006.

[7] Z. Qiongbing and D. Lixin, "A new crossover mechanism for genetic algorithms with variable-length chromosomes for path optimization problems," *Expert Systems with Applications*, vol. 60, pp. 183–189, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2016.04.005

[8] N. Hitomi and D. Selva, "Constellation optimization using an evolutionary algorithm with a variable-length chromosome," *IEEE Aerospace Conference Proceedings*, vol. 2018-March, pp. 1–12, 2018.

[9] J. T. Betts and W. P. Huffman, "Mesh refinement in direct transcription methods for optimal control," *Optimal Control Applications and Methods*, vol. 19, no. 1, pp. 1–21, 1998.

[10] S. Jain and P. Tsiotras, "Trajectory optimization using multiresolution techniques," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 5, pp. 1424–1436, 2008.

[11] J. Zhao and S. Li, "Modified multiresolution technique for mesh refinement in numerical optimal control," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 12, pp. 3324–3334, 2017.

[12] M. Vasile and F. Zuiani, "Multi-agent collaborative search: an agent-based memetic multi-objective optimization algorithm applied to space trajectory design," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 225, no. 11, pp. 1211–1227, 2011.

[13] G. R. Reeck, C. de Haen, D. C. Teller, R. F. Doollttle, W. M. Fitch, R. E. Dickerson, P. Chambon, A. D. McLachlan, E. Margoliash, T. H. Jukes, and E. Zuckerkandl, ""Homology" in Proteins and Nucleic Acids: A Terminology Muddle and a Way out of It," *Accountability in Research*, 1987.

[14] R. S. Zebulum, M. Vellasco, and M. A. Pacheco, "Variable length representation in evolutionary electronics," *Evolutionary Computation*, vol. 8, no. 1, pp. 93–120, 2000.

[15] W. R. Pearson, "An introduction to sequence similarity ("homology") searching," *Current protocols in bioinformatics*, vol. 42, no. 1, pp. 3–1, 2013.

[16] D. Goldberg, "Messy Genetic Algorithms : Motivation, Analysis, and First Results," *Complex Systems*, vol. 3, no. 5, pp. 493–530, 1989.

[17] D. S. Burke, K. A. De Jong, J. J. Grefenstette, C. L. Ramsey, and A. S. Wu, "Putting more genetics into genetic algorithms." *Evolutionary computation*, vol. 6, no. 4, pp. 387–410, 1998.

[18] B. Hutt and K. Warwick, "Synapsing variable-length crossover: Meaningful crossover for variable-length genomes," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 1, pp. 118–131, 2007.

[19] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "A review of optimization techniques in spacecraft flight trajectory design," *Progress in Aerospace Sciences*, vol. 109, no. June, p. 100543, 2019.

[20] J. T. Betts, *Practical methods for optimal control using nonlinear programming*. Society for Industrial and Applied Mathematics, 2001.

[21] C. L. Darby, W. W. Hager, and A. V. Rao, "An hp-adaptive pseudospectral method for solving optimal control problems," *Optimal Control Applications and Methods*, vol. 32, no. 4, pp. 476–502, 2011.

[22] M. Vasile, "Finite elements in time: a direct transcription method for optimal control problems," in *AIAA/AAS Astrodynamics Specialist Conference*, 2010, p. 8275.

[23] L. A. Ricciardi and M. Vasile, "Direct transcription of optimal control problems with finite elements on bernstein basis," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 2, pp. 229–243, 2019.

[24] L. A. Ricciardi, "Multi-Objective Hybrid Optimal Control with Application to Space Systems," Doctoral Dissertation, University of Strathclyde, 2019.

[25] O. Schutze, X. Esquivel, A. Lara, and C. A. C. Coello, "Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 4, pp. 504–522, 2012.

[26] J. Meditch, "On the problem of optimal thrust programming for a lunar soft landing," *IEEE Transactions on Automatic Control*, vol. 9, no. 4, pp. 477–484, 1964.

[27] L. A. Ricciardi and M. Vasile, "Improved archiving and search strategies for multi agent collaborative search," *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, pp. 435–455, 2019.