# Variable interactions and exploring parameter space in an expensive optimisation problem: optimising short term conflict alert

William J. Reckhouse, Jonathan E. Fieldsend *Member, IEEE* and Richard M. Everson *Member, IEEE*

***Abstract*— Short Term Conflict Alert (STCA) systems provide warnings to air traffic controllers if aircraft are in danger of becoming too close. They are complex software programs, with many inter-dependent parameters that must be adjusted to achieve the best trade-off between wanted and nuisance alerts. We describe a multi-archive evolutionary algorithm for optimising regional parameter subsets in parallel, reducing the number of evaluations required to generate an estimated Pareto optimal Receiver Operating Characteristic (ROC), showing that it provides superior results to traditional single-archived algorithms. A method of 'aggressive' optimisation, designed to explore unknown parameter ranges in a 'safe' manner, is shown to yield more extensive and better converged estimated Pareto fronts.**

## I. Introduction

NATS is the UK's main Air Navigation Service Provider, responsible for handling all 'en route' flights through UK airspace and providing Air Traffic Control services at 15 of the largest UK airports. NATS' business is focused on ensuring the safety of aircraft. The Short Term Conflict Alert (STCA) system is used at NATS to alert air traffic controllers to potential conflicts between aircraft which, in a worst-case scenario, might otherwise result in a dangerous loss of separation between the aircraft.

STCA systems are essentially classifiers that classify aircraft pairs into 'operationally relevant' encounters (for which an alert should be raised) and benign cases (for which an alert is not wanted) that would unnecessarily attract the controller's attention. Central to the safety case for STCA are two criteria: first, "the proportion of conflicts detected by the Controller in time for controller resolution will be enhanced by the use of STCA" and, second, "any negative effects on safety shall be small compared with the safety benefit and reduced as far as reasonably practical" [1]. The EUROCONTROL specification thus stipulates that an STCA system shall provide alerts for operationally relevant conflicts, while keeping nuisance alerts to an effective minimum.

The challenge for system analysts and maintenance personnel is in making informed decisions about a classifier's likely performance and selecting an appropriate trade-off between 'operationally relevant' alerts or true positives (TP) and those which are deemed nuisance alerts or false positives (FP). STCA systems are generally complex and tuning them manually involves adjustment of a great many (typically over 1000) operational parameters.

As airspace sectorisation can be complex STCA is set up to allow for 32 different *regions* (e.g. *en route*, approach, departure, stack, etc) in each of which the system's operation may be tailored to suit the behaviours of aircraft in that region. Exclusion zones may also be defined to cover areas outside a given control centre's authority or to mark uncontrolled airspace.

Parameters are set up to control how the system should react for each defined STCA region: there are multiple parameters defined for each of the system's modules and each parameter's value is specific to the region of airspace an aircraft pair is determined to be in. A *combined region type* is assigned to each aircraft pair based on which regions the constituent aircraft are in. This is done using a simple lookup matrix.

It is the large number of regions and resulting multiplicity of parameters that leads to the vast total number of STCA parameters that may be adjusted to affect the overall system performance. For example, there are ≈90 'regional' parameters in the ISS4 (issue 4 build) STCA implementation and 17 airspace regions defined for London Terminal Control Centre (LTCC), resulting in ≈1530 parameters overall.

In this paper we explore proposals for improving the efficacy of multi-objective optimisation algorithms for the automation of STCA tuning. We suggest methods of exploiting parameter subsets to aid convergence and implement a scheme for incrementally exploring unknown parameter space safely and efficiently.

## II. Single-archive optimisation

We regard the STCA system as a classifier $g(\mathbf{x};\boldsymbol{\theta})$ which gives an estimate of the probability that a feature vector $\mathbf{x}$ (for STCA the radar tracks of a pair of aircraft) belongs to one of two classes. We assume that the classifier depends upon a vector of adjustable parameters $\boldsymbol{\theta}$, and we denote by $TP(\boldsymbol{\theta})$ the classifier's true positive classification rate, while the false positive rate is denoted by $FP(\boldsymbol{\theta})$.

If the costs of an incorrect classification were known it would be straightforward to calculate the expected cost for any particular parameter and data set [2]. It would then be possible to adjust the parameters to minimise the expected cost. However, this procedure requires accurate specification of the misclassification costs which are seldom known accurately. The Receiver Operating Characteristic (ROC) curve is typically used to display the trade-off between true and false positive rates as the ratio of misclassification costs is varied for *soft* classifiers (see [3] for a recent review of ROC methods). For a *hard* classifier a single parametrisation ($\boldsymbol{\theta}$)

William Reckhouse, Jonathan Fieldsend and Richard Everson are with the College of Engineering, Mathematics and Physical Sciences, University of Exeter, EX4 4QF, UK. (email: {wjr202, J.E.Fieldsend, R.M.Everson}@exeter.ac.uk).

can only map to one point in ROC space. As such ROC curves for hard classifiers (of which NATS' STCA is one) can only be constructed by repeated evaluations of different parametrisations. With the ROC curves on hand the user can select the operating point with a full knowledge of the possible trade-offs involved.

In this section we show how multi-objective evolutionary algorithms may be used to derive the ROC curve for the STCA system optimised over all possible parameter values. That is, we seek to discover the set of parameters that simultaneously minimise $FP(\boldsymbol{\theta})$ and maximise $TP(\boldsymbol{\theta})$.

A general multi-objective optimisation problem seeks to simultaneously extremise $D$ objectives: $y_i = f_i(\boldsymbol{\theta})$, $i = 1, \ldots, D$, where each objective depends upon a vector $\boldsymbol{\theta}$ of $P$ parameters. It is convenient to assume that all the objectives are to be minimised, so for the STCA system we minimise the pair of objectives $(-TP(\boldsymbol{\theta}), FP(\boldsymbol{\theta}))$. The general multi-objective optimisation problem may be conveniently expressed as:

$$\text{minimise} \qquad \mathbf{y} = \mathbf{f}(\boldsymbol{\theta}) = (f_1(\boldsymbol{\theta}), \ldots, f_D(\boldsymbol{\theta})) \tag{1}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_P)$ and $\mathbf{y} = (y_1, \ldots, y_D)$. Note that the parameters may also be subject to some constraints, such as being within a safe operating range.

When there is more than one objective to be minimised parametrisations may exist for which performance on one objective cannot be improved without sacrificing performance on at least one other. Such parametrisations are said to be *Pareto optimal* [4], [5], [6], [7] and the set of all Pareto optimal parametrisations is said to form the Pareto set.

The notion of *dominance* is often used to make Pareto optimality clearer. $\boldsymbol{\theta}$ is said to *strictly dominate* $\boldsymbol{\phi}$ (denoted $\boldsymbol{\theta} \prec \boldsymbol{\phi}$) if and only if all the objectives corresponding to the parameter $\boldsymbol{\theta}$ are no worse than those obtained with $\boldsymbol{\phi}$, and at least one objective is better:

$$\begin{aligned} f_i(\boldsymbol{\theta}) \leq f_i(\boldsymbol{\phi}) \quad \forall i = 1, \ldots, D \quad \text{and} \\ f_i(\boldsymbol{\theta}) < f_i(\boldsymbol{\phi}) \quad \text{for some} \quad i. \end{aligned} \tag{2}$$

A set $A$ of decision vectors is said to be a *non-dominated set* if no member of the set is dominated by any other member:

$$\boldsymbol{\theta} \not\prec \boldsymbol{\phi} \quad \text{for all } \boldsymbol{\theta}, \boldsymbol{\phi} \in A. \tag{3}$$

A solution to the minimisation problem (1) is thus *Pareto optimal* if it is not dominated by any other feasible parametrisation. The image of the non-dominated set of all Pareto optimal parametrisations in objective space is known as the *Pareto front*. Recent years have seen the development of a number of evolutionary techniques based on dominance measures for locating the Pareto front; see [8], [9], [4], [6], [7] for recent reviews. We have previously implemented a simple but effective iterative approach for the STCA optimisation problem, in which repeated perturbation and evaluation for dominance leads a population of increasingly fit parametrisations toward the Pareto-optimal front [10].

Algorithm 1 outlines the procedure used in this earlier work which is the the basis of the extended multi-archive model described here. At line 1 the archive, $A$, is initialised (usually by generating a set of parametrisations each with randomised values, or seeding with a known 'good' parametrisation). An initial set of parametrisations is evaluated on the STCA system and added to the archive depending on dominance (see algorithm 2, `update`). At line 3 a parent parametrisation, $\boldsymbol{\theta}$, is selected from archive $A$, either at random or with some biased scheme such as uniselect[1] The parent parametrisation is then perturbed to create a new child parametrisation, $\boldsymbol{\theta}'$, at line 4. At line 5, the child parametrisation, $\boldsymbol{\theta}'$, is evaluated on the STCA system, its associated true and false positive rates are obtained. The child parametrisation is then updated to the archive $A$ at line 6. The algorithm repeats the process $N$ times from line 2.

**Algorithm 1** Single-archive `Main()`

```
1: $A :=$ initialise()
2: for $n := 1$ to $N$ do
3:     $\boldsymbol{\theta} :=$ select($A$)
4:     $\boldsymbol{\theta}' :=$ perturb($\boldsymbol{\theta}$)
5:     $(TP(\boldsymbol{\theta}'), FP(\boldsymbol{\theta}')) := STCA(\boldsymbol{\theta}')$
6:     update($A, \boldsymbol{\theta}'$)
```

**Algorithm 2** `update`$(A, \boldsymbol{\theta}')$

```
1: if $\nexists(\boldsymbol{\phi} \in A \text{ s.t. } \boldsymbol{\phi} \preceq \boldsymbol{\theta}')$ then
2:     $A := \{\boldsymbol{\phi} \in A \,|\, \boldsymbol{\theta}' \not\preceq \boldsymbol{\phi}\}$
3:     $A := A \cup \{\boldsymbol{\theta}'\}$
```

Algorithm 2 outlines the `update` method used to determine whether the new parametrisation, $\boldsymbol{\theta}'$, should be added to archive $A$. If the new parametrisation is not dominated by any existing members of the archive (line 1), then dominated members of $A$ are removed (line 2) and the new parametrisation, $\boldsymbol{\theta}'$, is added to $A$ (line 3). Otherwise the new parametrisation is discarded.

It was found that algorithm 1 could, in a few thousand iterations, find parametrisations dominating the NATS manually tuned Current Operating Point (COP) when initialised from a random set of solutions, as well as locate a good spread of trade-off solutions for staff at NATS to consider. However due to the expense of the problem (time per evaluation), runs would take between a fortnight and a month (≈3m 30s CPU time per evaluation on ≈70000 aircraft pairs, STCA ISS4). As such the novel work below develops a more *domain specific* optimiser, leveraging the semi-independent nature of sections of the system parametrisation. The work below should also prove beneficial to tackling other industrial problems whose decision vectors are similarly structured.

## III. Multi-archive optimisation

The off-line STCA model is capable of reporting STCA performance on a region by region basis. If we were to

[1] Uniselect [11] encourages the 'filling-out' or spreading of points across the ROC by biasing selection towards those members in a more sparsely populated region of the curve. A point is chosen at random (uniformly on one objective coordinate) and nearest archive member returned for perturbation; the objective on which selection is performed is swapped each iteration.

**Algorithm 3** $\texttt{compositeSelect}(A, \{A_i \mid i = 1, \ldots, R\})$

```
1: $\boldsymbol{\theta} := \texttt{select}(A)$
2: $\boldsymbol{\theta}' := \texttt{nonRegionalParams}(\boldsymbol{\theta})$
3: for $i := 1$ to $R$ do
4:     $\boldsymbol{\phi} := \texttt{select}(A_i)$
5:     $\boldsymbol{\theta}' := \texttt{append}(\boldsymbol{\theta}', \boldsymbol{\phi})$
6: return $\boldsymbol{\theta}'$
```

**Algorithm 4** $\texttt{compositeUpdate}(\boldsymbol{\theta}, A, \{A_i \mid i = 1, \ldots, R\})$

```
1: $\texttt{update}(A, \boldsymbol{\theta})$
2: for $i := 1$ to $R$ do
3:     $\boldsymbol{\phi} := \texttt{extractRegion}(\boldsymbol{\theta}, i)$
4:     $\texttt{update}(A_i, \boldsymbol{\phi})$
```

assume that each region's corresponding parameters are independent of other regions, then optimisation of the regional subsets could be performed separately. Evaluations for all regions could be executed in parallel by splicing regional parameters into a single parametrisation for simultaneous evaluation on STCA. If $R$ is the number of regions in the parameter set, we perform $R$ simultaneous optimisations on each regional parameter subset, as well as maintaining an 'overall' archive that depends upon the dominance of each combined parametrisation. New parametrisations generated by the optimiser are a composite of the non-dominated subsets found in all the other archives. The subsequent splicing of regional subsets to form a child parametrisation could be thought of as a form of crossover in genotype.

In addition to the regional parameters, STCA has a small number (under a dozen) of non-regional parameters, which apply to the modules across all regions. Non-regional parameters are selected from the 'overall' archive and combined with the regional subsets to complete the new parameter set for evaluation.

The outline of the multi-archive algorithm remains largely unchanged from that of algorithm 1, however the `select` and `update` methods have been replaced with `compositeSelect` and `compositeUpdate` respectively.

The `compositeSelect` method described in algorithm 3 is used to combine parameter subsets from all the regional archives $A_i$ ($i = 1, \ldots, R$) to create a new composite parametrisation. Selection of parameter subsets from the archives can either be uniform or based on the uniselect method described in [11].

At line 1 a member, $\boldsymbol{\theta}$, of the overall archive $A$ is selected. At line 2 the non-regional parameters subset is copied from $\boldsymbol{\theta}$ and assigned to $\boldsymbol{\theta}'$. Lines 3 to 5 concern the selection and splicing of regional parameter subsets from $\{A_i\}$. At line 4 a regional parameter subset, $\boldsymbol{\phi}$, is selected from regional archive $A_i$. The selected subset, $\boldsymbol{\phi}$, is appended to $\boldsymbol{\theta}'$ at line 5. After all regional subsets have been appended the composite parametrisation, $\boldsymbol{\theta}'$, is returned (line 6).

Algorithm 4 outlines the multi-archive update procedure. At line 1 the overall archive, $A$, is updated with $\boldsymbol{\theta}$ using the parametrisations overall TP and FP rates. At line 3 a subset of parameters, $\boldsymbol{\phi}$, corresponding to region $i$, is extracted from $\boldsymbol{\theta}$. The regional parameters subset, $\boldsymbol{\phi}$, is then updated to the corresponding regional archive $A_i$ at line 4. Dominance is determined based on the subset's regional TP and FP rates, not the overall parametrisation's rates.

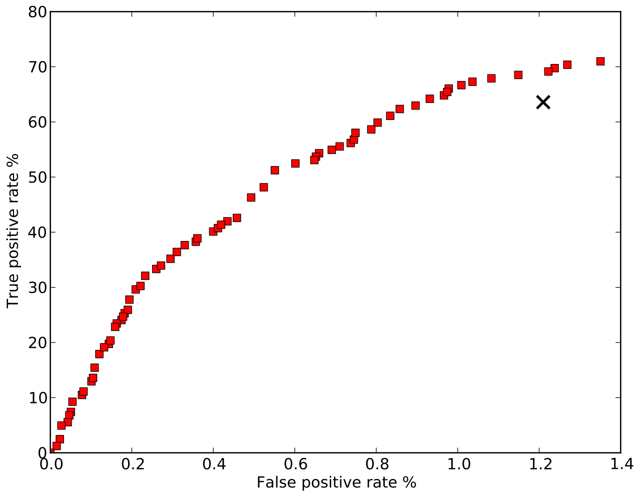

Fig. 1. Multi-archive results (6468 iterations).

### *A. Results*

Figure 1 shows the archive contents after 6468 iterations using ≈70,000 aircraft pairs from the Manchester Area Control Centre (MACC) during July 2007. The optimiser has located an ROC consisting of 68 points ranging from 0.00% to 70.99% TP and 0.00% to 1.35% FP. The NATS manually tuned COP was not included in the optimiser's initial set, however it has been marked on the chart for comparison (a black cross in figure 1, 63.58% TP and 1.21% FP). The initial set comprised 100 parametrisations, each initialised with random parameter values.

The optimiser has located points that entirely dominate (are wholly better than) the COP. That is, for approximately the same tolerated FP rate the optimiser has located a parametrisation ≈5.6% better in TP rate than the COP; for approximately the same tolerated TP rate the optimiser has located a parametrisation ≈0.3% better in FP rate than the COP. Although the improvement over the COP is relatively small in percentage terms, the quantity of aircraft track pairs processed by STCA means that a significant reduction in the number of false alerts could be achieved while maintaining the current wanted alert rate. We note that the COP is derived from over a decade of manual tuning and also includes optimisation for the warning time which means that the NATS parameters cannot be immediately changed to a more optimal set from these results alone.

In addition, perhaps of greater significance, the multi-archive optimiser has located a better ROC in fewer than half the number of iterations that were required by single-archive mode. Figure 4 shows an overlay of the ROCs generated by single (blue circles) and multi (red squares) archive modes

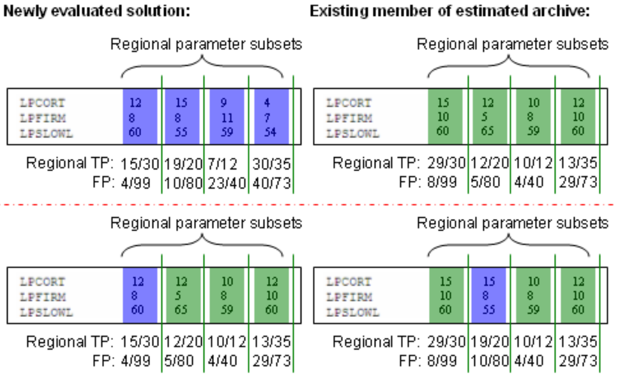
Newly evaluated solution:
Existing member of estimated archive:
Regional parameter subsets
Regional parameter subsets
LPCORT
LPFIRM
LPSLOWL
12 8 60
15 8 55
9 11 59
4 7 54
LPCORT
LPFIRM
LPSLOWL
15 10 60
12 5 65
10 8 59
12 10 60
Regional TP: 15/30 19/20 7/12 30/35
FP: 4/99 10/80 23/40 40/73
Regional TP: 29/30 12/20 10/12 13/35
FP: 8/99 5/80 4/40 29/73
Regional parameter subsets
Regional parameter subsets
LPCORT
LPFIRM
LPSLOWL
12 8 60
12 5 65
10 8 59
12 10 60
LPCORT
LPFIRM
LPSLOWL
15 10 60
15 8 55
10 8 59
12 10 60
Regional TP: 15/30 12/20 10/12 13/35
FP: 4/99 5/80 4/40 29/73
Regional TP: 29/30 19/20 10/12 13/35
FP: 8/99 10/80 4/40 29/73

Fig. 2. Creation of estimated parametrisations.

after 6468 iterations. Even after a further 7254 iterations the single archive mode has still not caught up with the better TP and FP rates located by the multi-archive optimiser.

## IV. Estimated-archive optimisation

In the this section we expand on the multi-archive concept with the aim of further reducing the number of iterations required for optimal ROC generation. A method of recombining evaluated parameter subsets based on estimated TP and TP rates is presented.

The multi-archive mode is possible because we can obtain the number of TP and FP alerts found in each region of airspace, not just the overall STCA alerts rate, permitting optimisation on regional subsets of the STCA parameters. However, the parameter subsets present in the regional archives can be recombined in many different ways, resulting in entirely new parametrisations. Estimated TP and FP counts for these new parametrisations can be calculated by adding together the associated regional TP and FP counts from the constituent subsets in the regional archives. The estimated-archive should therefore represent the estimated 'fitness' of non-dominated parametrisations resulting from all possible recombinations of regional parameter subsets.

Of course generating every single combination of parameter subset is non-trivial for any reasonably large set of parametrisations. We can achieve a good approximation by simply substituting each regional subset of a newly evaluated parametrisation with the corresponding subset from each parametrisation already existing in the estimated archive. The new parametrisation may also be added to the estimated archive if it is not dominated by the existing members.

If the subsets of regional STCA parameters were truly independent, then the estimated TP and FP rates of each recombined parametrisation would be the same as if the parametrisations were evaluated for real on the STCA system. The estimated archive provides a means of generating hundreds of, potentially better, parametrisations without the need for evaluating each one. It might therefore be expected to increase the speed of optimisation runs by reducing the number of evaluations necessary.

Figure 2 illustrates the partial recombination procedure. The top row depicts a newly evaluated parametrisation (TP: 71/97 = 73%, FP: 77/292 = 26%) along side an existing member of the estimated archive (TP: 64/97 = 66%, FP: 46/292 = 16%). Below are two possible combinations formed by inserting regional parameter subsets from the newly evaluated parametrisation into the estimated parametrisation. Estimated TP and FP rates are calculated by summing the associated regional TP and FP counts; TP: 50/97 = 52%, FP: 42/292 = 14% for the left hand parametrisation, TP: 71/97 = 54%, FP: 51/292 = 17% for the right. In reality however, there are more than four STCA regions to recombine.

**Algorithm 5** Estimated-archive Main()

```
1:  $I :=$ initialise()
2:  $A := \emptyset$
3:  $E := \emptyset$
4:  for $\boldsymbol{\theta}$ in $I$ do
5:      update$(A, \boldsymbol{\theta})$
6:      updateEstimates$(E, \boldsymbol{\theta})$
7:  for $n := 1$ to $N$ do
8:      $\boldsymbol{\theta} :=$ select$(E)$
9:      $\boldsymbol{\theta}' :=$ perturb$(\boldsymbol{\theta})$
10:     $(TP(\boldsymbol{\theta}'), FP(\boldsymbol{\theta}')) := STCA(\boldsymbol{\theta}')$
11:     update$(A, \boldsymbol{\theta}')$
12:     updateEstimates$(E, \boldsymbol{\theta}')$
```

Algorithm 5 outlines the estimated-archive procedure more formally. At line 1 a set of parametrisations, $I$, is initialised and each member evaluated on the STCA system. However, dominance of the members of $I$ is not determined at this stage. Some of the parametrisations may therefore dominate others. Lines 2 and 3 initialise the 'overall' archive, $A$, and 'estimated' archive, $E$, as empty sets. Lines 4 to 6 update each of the archives with each parametrisation, $\boldsymbol{\theta}$, in set $I$. Note at line 6 a new method, 'updateEstimates', is used to update the estimated archive (see algorithm 6). The archives have now been initialised.

The evolutionary optimisation loop is entered at line 7. At line 8 a parent parametrisation, $\boldsymbol{\theta}$, is selected from the *estimated* archive, $E$, using either uniform or uniselect selection. The parent parametrisation is then perturbed to create a new child parametrisation, $\boldsymbol{\theta}'$, at line 9. Perturbation must ensure that at least one parameter from each regional subset is changed. At line 10 the child parametrisation, $\boldsymbol{\theta}'$, is evaluated on the STCA system. At line 11 the overall archive, $A$, is updated with $\boldsymbol{\theta}'$ and at line 12 the estimated archive, $E$, is updated using the updateEstimates method.

The updateEstimates method described in algorithm 6 is used to generate multiple estimated parametrisations by combining each existing member of the estimated archive, $E$, with newly evaluated parametrisation $\boldsymbol{\theta}'$. The estimated archive is updated with each new estimated parametrisation; this means that for every evaluated parametrisation, $\boldsymbol{\theta}'$, archive $E$ will be updated with multiple estimates.

At line 1 the algorithm begins to iterate through each existing member, $\boldsymbol{\theta}$, of estimated archive $E$. Lines 2 and 4 form a nested loop designed to combine regional parameter

**Algorithm 6** `updateEstimates`$(E, \boldsymbol{\theta}')$

```
1:  for θ in E do
2:      for i := 1 to R do
3:          φ := nonRegionalParameters(getParent(θ))
4:          for j := 1 to R do
5:              if i = j then
6:                  χ := extractRegion(θ′, i)
7:              else
8:                  χ := extractRegion(θ, j)
9:              φ := append(φ, χ)
10:         update(E, φ)
11: update(E, θ′)
```

subsets from $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. In order to maintain diversity in non-regional parameters `getParent` selects one of the parents of the parametrisation $\boldsymbol{\theta}$. Line 3 assigns parametrisation $\boldsymbol{\phi}$ 'non-regional' parameters selected in a uniform manner (using `getParent`) from the set of parent parametrisations belonging to $\boldsymbol{\theta}$. Note that the non-regional parameters of $\boldsymbol{\theta}$ itself may also be selected. At line 6 if region $i$ equals region $j$ then the parameter subset corresponding to region $i$ from parametrisation $\boldsymbol{\theta}'$ is assigned to $\boldsymbol{\chi}$; otherwise the parameter subset corresponding to region $j$ from parametrisation $\boldsymbol{\theta}$ is assigned to $\boldsymbol{\chi}$ (line 8). $\boldsymbol{\chi}$ is then appended to parametrisation $\boldsymbol{\phi}$ (line 9). Once estimated parametrisation $\boldsymbol{\phi}$ has a complete set of regional parameters it is updated to the estimated archive, $E$ (line 10). The process of combining regional parameter subsets is then repeated for the next region. When iteration through each existing member of $E$ has completed, parametrisation $\boldsymbol{\theta}'$ itself is updated to archive $E$ (line 11).

### A. *Results*

Figure 3 shows the archives' contents after 12677 iterations using the same MACC data as section III. The estimating optimiser has located an actual (evaluated) ROC consisting of 46 solutions ranging from 0.00% to 57.41% TP and 0.00% to 1.16% FP (yellow diamonds). An 'estimated' ROC consisting of 77 points, lies from 9.88% to 62.35% TP, 0.00% to 1.08% FP (purple diamonds). The run was initialised with 100 randomised parametrisations. In this case the NATS' COP (marked by the black cross) has not been dominated (the optimiser has not located any points that are wholly better than the manually tuned parametrisation).

Even after 12677 iterations the estimated archive still lies ahead of the actual 'overall' archive of *evaluated* parametrisations. If the estimated archive were working correctly it is expected that the 'overall' archive would eventually converge upon the estimated archive's position as the optimal ROC is located. This however, has not occurred; while the lack of convergence may imply that the optimiser simply has not run for long enough, further analysis suggests otherwise.

Comparing the evaluated front (yellow diamonds) to those generated by previous trials (figure 4), we see that it has largely dominated the single-archive scheme and has managed to dominate the multi-archive scheme in one place

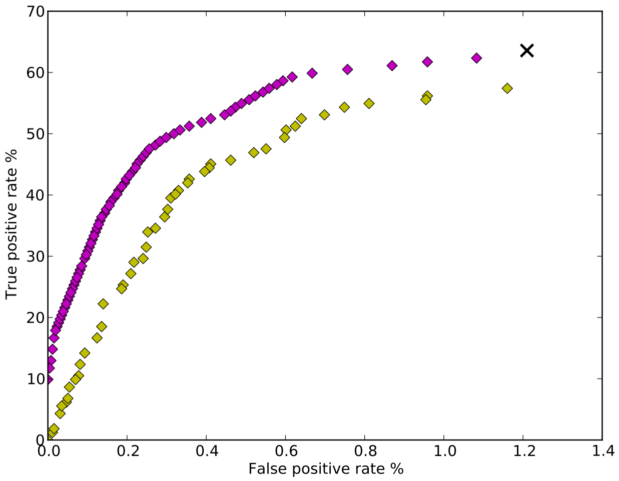

Fig. 3. Estimated-archive results (12677 iterations).

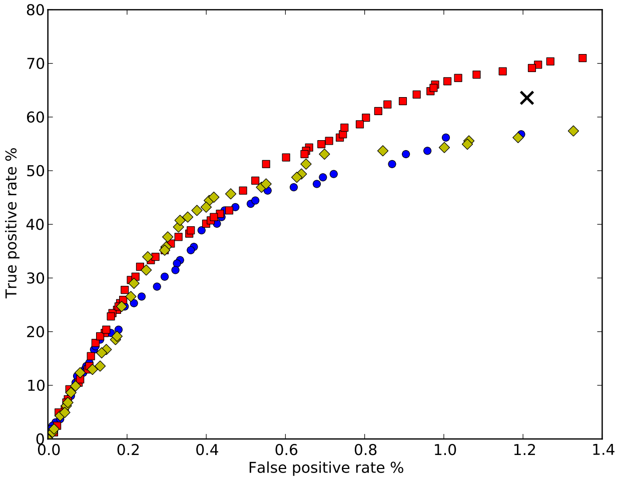

Fig. 4. Comparison of single (blue circles), multi- (red squares) and estimated (yellow diamonds) archive optimisation schemes after 6468 iterations.

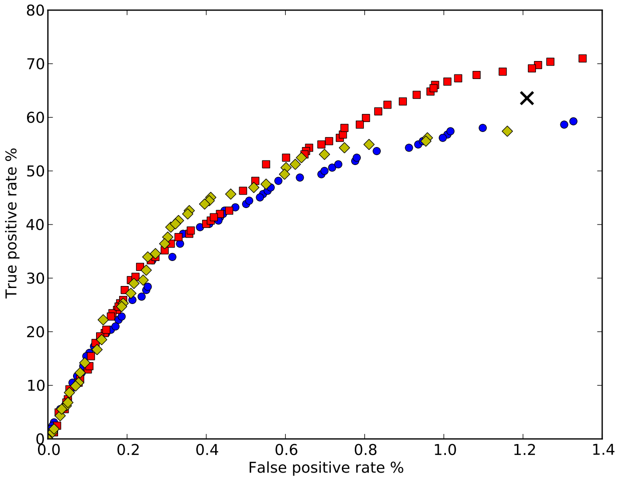

Fig. 5. Comparison of single (blue circles), multi- (red squares) and estimated (yellow diamonds) archive optimisation schemes after 12677 iterations for the single and estimated schemes (6209 iterations in addition to figure 4); the multi-archive front is at 6468 iterations as in figure 4.

(≈36% to ≈45% TP, ≈0.3% to ≈0.45% FP). Referring to figure 5 we see that the estimated archive algorithm's evaluated front still dominates single archive's front, however single archive has begun to catch up. The dominance over multi-archive mode has not greatly improved.

The estimated archive algorithm's evaluated front has not really progressed in the subsequent 6209 iterations; it seems to have stalled. This indicates that although in early iterations it has a speed advantage over single archive mode (converging faster), the estimated archive appears to eventually inhibit further progress. It should also be noted that, if the multi-archive algorithm were to be allowed to run for longer it is likely that it too would locate the area of the ROC currently dominated by the estimated-archive's evaluated front.

This eventual inhibition of the estimated-archive is, we think, caused by a greater degree of interaction between regional parameters than was previously anticipated. This arises because aircraft pairs with aircraft in different regions must be assigned to a single region by STCA; as regional parameters vary during the optimisation, aircraft pair assignment shifts between regions, inducing a dependency between these regions. In addition the (relatively few) non-regional parameters have been found to have a large effect. Dependencies between regional parameters introduces noise reducing the effectiveness of parametrisation perturbation. As the optimisation progresses the estimated fitness of parametrisations selected from the estimated archive (the purple diamonds in figure 3) are so incongruent with the actual evaluated fitnesses that new parametrisations entering the evaluated archive (the yellow diamonds) are always found to be dominated by existing members. Exploration of the search space is inhibited as the 'gene-pool' stagnates due to selection from 'corrupted' parent members in the estimated archive. Future investigation of co-evolutionary schemes [12] may present a solution to the problem of parameter dependencies and allow exploration of decision variable space in a more general way.

## V. Aggressive optimisation

The trials presented so far have been 'conservative', that is perturbation of parameters was restricted to ranges previously used by NATS at the Current Operating Point. Conservative values were determined by analysing all previously used STCA settings (over the past 12 years) for each parameter. This was done as a validation exercise, ensuring that the optimiser could reach (and exceed) the COP without explicit knowledge of it. It also limited potential problems associated with unknown parameter ranges causing STCA system instability, providing an assurance that the optimised system was still operating within the usual parameter ranges. We now describe an 'aggressive' algorithm designed to explore the unknown parameter ranges in a 'safe' manner.

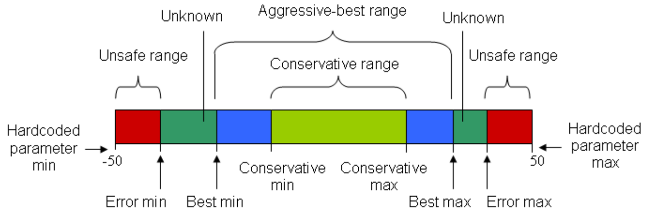

Fig. 6. Schematic diagram of conservative and aggressive parameter ranges.

STCA has a hardcoded range specified for each parameter, these were defined during its development. It should be noted that due to the complexity of STCA, there are combinations of parameters, *within* the hardcoded ranges, that can be rejected by the system. In addition, due to coupling between the parameters, it is not possible to simply find the safe range for one parameter alone, as its outcome depends upon the value of others in the parametrisation as a whole. Therefore, as a result of the interdependence, individual searches on each parameter cannot be performed; the exploration of unknown ranges must be implemented as part of the optimisation process itself.

The process of determining safe parameter boundaries is inherently tied into the optimisation algorithm; implementing optimisation at the same time as expanding parameter ranges ensures that only relevant boundary modifications are made. That is, there is no point in expanding ranges that do not result in a better performance, so only those ranges restricting relevant search space exploration should be modified.

The basic principle is to increase a parameter's maximum range until the STCA model becomes unstable (rejects an evaluation). For each parameter eight values are tracked (figure 6): the conservative or 'safe' min/max, the 'best'/most exploratory values achieved so far, the min/max values that last resulted in an STCA rejection, and the STCA hardcoded min/max range as defined in the STCA specification. At initialisation the 'best'/most exploratory values are set equal to the conservative or 'safe' range; the values that 'last resulted in an STCA rejection' will be set to the hardcoded range.

Each time a parametrisation is selected for perturbation we permit only one of the parameters selected for mutation to be modified outside the 'best'/most exploratory range reached so far; all other parameters are mutated within previously explored ranges. If STCA accepts the parametrisation the 'best' so far value is updated; should a rejection occur the 'error max' or 'error min' value for the parameter is updated.

The method of mutation (i.e. random incrementation/decrementation based on Laplacian/Gaussian probability distribution) remains unchanged. Note that this means that just because a parameter may be modified outside of the 'best'/most exploratory range so far, perturbation does not necessarily guarantee that it will be. This ensures that parameter values closest to bounds are more likely extend the range, whilst those parameters whose current value is in a more conservative range tend to leave the boundaries unchanged.

It is straightforward to amend the algorithms previously defined for all other optimisation schemes to include aggressive parameter perturbation. The required modifications are as follows:

- An additional method '`updateRanges`' is called when parametrisation evaluations complete.

- The ‘perturb’ method responsible for parameter value perturbation is permitted to use the ‘aggressive-best’ range defined for a given parameter (which is initially equal to the ‘conservative’ range). Values may be perturbed outside of the ‘aggressive-best’ range once per parametrisation, after which all subsequent perturbations performed on the parametrisation are constrained to lie *within* the ‘aggressive-best’ range.

### A. Results

Figure 7 shows the multi-archive, aggressive, ‘overall’ archive’s contents after 11900 iterations on MACC data. The optimiser has located an ROC consisting of 60 points ranging from 3.52% to 93.47% TP and 0.00% to 14.35% FP. The trial was initialised with 100 randomised parametrisations.

The aggressive optimisation run (red circles) is compared to a previous ‘conservative’ trial on the same data (blue squares). At low TP and FP rates the conservative and aggressive fronts virtually coincide. The additional freedom allowed to the parameters in the aggressive case permits new solutions at higher TP and FP rates to be discovered, resulting in a more complete picture of the Pareto optimal curve. The COP is now observed to be close to the corner or ‘knee’ of the ROC and the optimiser has located operating points with performance well beyond that of the COP. For approximately the same tolerated FP rate the optimiser has located a parametrisation ≈7.5% better in TP rate than the COP, while for approximately the same tolerated TP rate the optimiser has located a parametrisation ≈0.5% better in FP rate than the COP.

Use of the aggressive optimisation scheme has little or no impact on optimisation speed when compared to the equivalent conservative trial. That is, although there are some wasted perturbations due to evaluation failures, these are more than made up for by the overall improvement in parametrisation quality. Thus, the aggressive algorithm produces a better quality ROC without having a negative impact on the time taken to generate it. Additionally, by analysing those parameter settings that cause STCA to reject, the optimiser can highlight instances of incorrectly defined variable ranges.

Figure 8 illustrates the extent to which some parameter ranges have been extended. The charts have been normalised to each parameter’s min/max conservative range (marked in green 0.0-1.0). The black points represent how far the optimiser has extended the aggressive-best range into unknown parameter space, blue points mark each parameter’s current value in the given parametrisation. The figures show the normalised ranges for the parametrisation which directly dominates the NATS COP in figure 7 (the corner or ‘knee’ of the ROC; 68.84% TP, 1.03% FP). The right-hand plot is a duplicate of the left-hand side, however, it has not been cropped to highlight smaller values and thus shows the true extent to which some ranges have been extended. Note the significant number of parameter values (marked in blue) that lie outside of the conservative range (marked in green).

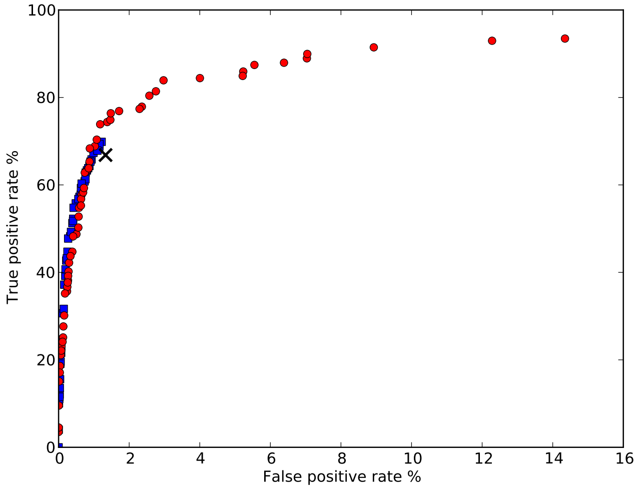

Fig. 7. ‘Aggressive’ optimisation (red circles) results compared with conservative optimisation (blue squares) after 11900 iterations.

## VI. Results summary

TABLE I

| Mode | Front size | AUC | Dist. to RAL | TP range (%) | FP range (%) |
|---|---|---|---|---|---|
| Single | 48 | 0.57 | 0.39 | 0.62-56.79 | 0.00-1.20 |
| Multi | 68 | 0.71 | 0.49 | 0.00-70.99 | 0.00-1.35 |
| Estimated | 44 | 0.57 | 0.40 | 0.00-57.41 | 0.00-1.33 |
| Conservative | 48 | 0.69 | 0.48 | 0.00-69.35 | 0.00-1.28 |
| Aggressive | 61 | 0.92 | 0.58 | 3.52-93.47 | 0.00-14.35 |

A brief summary of results from the preceding trials is included in table I. For the purposes of comparison all figures were obtained at 6468 iterations. Note that the conservative and aggressive trials were run on a modified dataset and thus are not directly comparable to the other results.

Taking the area under the ROC (AUC) and the maximum distance from the random allocation line (drawn between 0, 0 and 100, 100) as measures of convergence we can clearly see that the multi-archive optimiser outperforms single and estimated-archive modes in terms of speed of convergence, number of parametrisations, and range of TP/FP rates covered. The single and estimated-archive modes are seen to be roughly equivalent in performance. Aggressive optimisation shows a marked improvement over conservative, permitting a much wider range of TP/FP rates to be covered by the Pareto front.

## VII. Conclusions

The speed of solution evaluations on the off-line STCA model is a major limiting factor in this optimisation scenario. While parallelisation could be suggested as a possible solution, it would seek to improve on speed by simply increasing the number of evaluations possible in a given period (tapping extra processing power). The multi-archive optimiser instead seeks to improve optimisation speed by *reducing* the number of evaluations that are *necessary* to generate an ROC curve of equal progression.

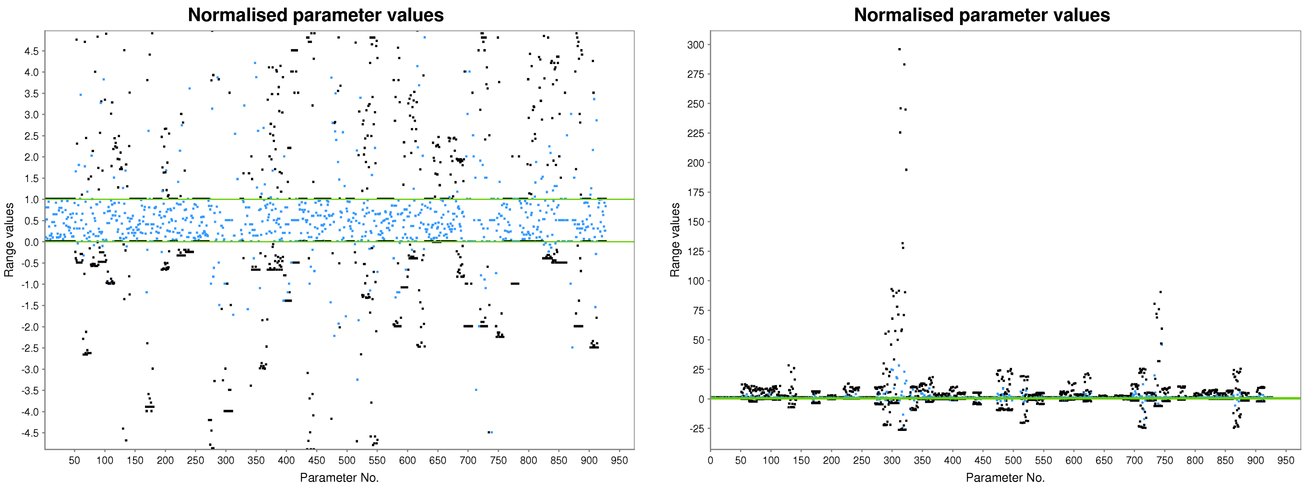

Fig. 8. Parameter ranges plot normalised to the 'conservative' range.

We have shown how maintaining a set of elite archives, one for each regional parameter subset, can significantly reduce the number of evaluations required to generate a good estimate of the Pareto-optimal ROC for the NATS STCA system, thus increasing the speed of optimisation. In addition, the multi-archive approach can generate an ROC which entirely dominates that located by the single-archive optimiser (although at lower TP and FP rates the fronts virtually coincide).

However, the multi-archive approach assumes regional parameter independence. In reality this is not the case due to non-regional parameters, cross-regional aircraft encounters and dataset region labelling discrepancies. This effectively means that some noise is introduced to the candidate parametrisations, reducing the effectiveness of the optimisation approach.

The effects of parameter dependency are most pronounced when using the estimated-archive; although initial progression of the ROC is encouraging, further advance of the front appears to stall after only a few thousand iterations. At low TP and FP rates the ROC is roughly equivalent to both single and multi-archive algorithms, however the estimated-archive scheme fails to dominate multi-archive's front at higher TP and FP rates, even after it has been run for double the number of iterations. We speculate that noise, introduced by dependencies between regional parameters, inhibits the estimated-archive's development.

Our method of incrementally exploring 'unknown' regions of parameter space in a 'safe' and efficient manner, has shown that a more complete picture of the ROC curve may be obtained without increasing the number of iterations.

'Multi-archive' mode with aggressive perturbation is by far the most effective means of optimising the STCA system that we have investigated thus far, and we anticipate that the methods developed here will be generally applicable to industrial optimisation problems.

## Acknowledgment

This work was supported by a Knowledge Transfer Partnership awarded to NATS and the University of Exeter. Special thanks to David Bush, Trevor Arnold, Richard Hayward and Keith Slater from the Operational Analysis Department, NATS.

## References

[1] Eurocontrol, "Specification for short term conflict alert (edition 1.0), Nov. 2007." [Online]. Available: www.eurocontrol.int/safety-nets/gallery/content/public/es_STCA-10.pdf

[2] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[3] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, pp. 861–874, 2006.

[4] C. Coello, "A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques," *Knowledge and Information Systems. An International Journal*, vol. 1, no. 3, pp. 269–308, 1999.

[5] D. V. Veldhuizen and G. Lamont, "Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art," *Evolutionary Computation*, vol. 8, no. 2, pp. 125–147, 2000.

[6] K. Tan, T. Lee, and E. Khor, "Evolutionary Algorithms for Multi-objective Optimization: Performance Assessments and Comparisons," *Artificial Intelligence Review*, no. 17, pp. 253–260, 2002.

[7] C. Fonseca and P. Fleming, "An Overview of Evolutionary Algorithms in Multiobjective Optimization," *Evolutionary Computation*, vol. 3, no. 1, pp. 1–16, 1995.

[8] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester: Wiley, 2001.

[9] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. Grunert da Fonseca, "Performance Assessment of Multiobjective Optimizers: An Analysis and Review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[10] R. Everson and J. Fieldsend, "Multi-Objective Optimisation of Safety Related Systems: An Application to Short Term Conflict Alert," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 2, pp. 187–198, 2006.

[11] K. I. Smith, "A Study of Simulated Annealing Techniques for Multi-Objective Optimisation," Ph.D. dissertation, University of Exeter, 2006.

[12] C.-K. Goh and K. C. Tan, "A Competitive-Cooperative Coevolutionary Paradigm for Dynamic Multiobjective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 103–127, February 2009.