MOPC/D: A new Probability Collectives algorithm for Multiobjective Optimisation

David Morgan
BAE Systems
Advanced Technology Centre
Filton, UK
Email: david.morgan11@baesystems.com

Antony Waldock
BAE Systems
Advanced Technology Centre
Filton, UK
Email: antony.waldock@baesystems.com

David Corne
School of MACS
Heriot-Watt University
Edinburgh, UK
Email: dwcorne@macs.hw.ac.uk

***Abstract*—Decomposition strategies in Multiobjective optimisation (MOO) are known to be superior to other approaches on a wide variety of problems. Probability Collectives (PC) is a recent distribution-centric optimisation framework that has origins in game-theory and statistical physics. Here, we present a new Probability Collectives MOO algorithm, MOPC/D, based on a decomposition strategy that exploits the search operators which arise naturally from the use of a probabilistic Gaussian mixture model formulation. Evaluation of this approach, using the 2- and 3- objective unconstrained problems from the CEC2009 benchmark suite, found MOPC/D to perform competitively with the state of the art (across these problems it has the best mean rank and rank standard deviation of 14 algorithms in the CEC2009 competition, e.g. above MOEA/D), and significantly outperform the (only) previously published MOO algorithm in the PC framework. We conclude that the performance of MOPC/D shows considerable promise, and suggest a number of lines for further research.**

I. Introduction

Many real-world problems require simultaneous optimisation of a decision vector $\mathbf{x}$ with respect to several different objective measures. These often conflict, such that small changes $\mathbf{x} \rightarrow \mathbf{x}+\delta\mathbf{x}$ which improve performance with respect to some objectives usually degrade performance with respect to others. Most multi-objective problems possess no single solution that is simultaneously optimal with respect to all objectives, so instead the desired solution is the set of Pareto optimal solutions $\{\mathbf{x}^*\}$ that represent the best achievable trade-off. Consider the class of continuous multi-objective problems defined by the set of objective functions $G_j : \mathbf{x} \rightarrow y_j\,,\ j \in \{1,...,b\}$ which map points $\mathbf{x} \in \Omega_1 \subseteq \mathbb{R}^a$ in a known a-dimensional decision space Ω_1 into points $\mathbf{y} \in \Omega_2 \subseteq \mathbb{R}^b$ in a (generally) unknown b-dimensional objective space Ω_2. The Pareto set for this problem contains all vectors $\mathbf{x}^*$ whose associated objective vector $\mathbf{y}$ is non-dominated as defined by (1).

$$\min_j \left(G_j\left(\mathbf{x}^*\right) - G_j\left(\mathbf{x}\right)\right) < 0 \ \forall\ \mathbf{x} \in \Omega_1 \tag{1}$$

Obtaining the Pareto set for a problem in this class is typically NP-Hard or NP-Complete. Additionally, many real-world problems involve objective functions that are themselves driven by complex models, so the primary feature of an effective MOO metaheuristic is an ability to generate high quality Pareto set approximations whilst expending relatively few function evaluations. To date, the most successful and widely adopted approach has been multiobjective evolutionary algorithms (MOEAs), which evolve a population of solution vectors using genetic operators and selection routines according to one or more fitness functions. Most commonly these are based around Pareto dominance [8], but approaches based around aggregation [2] and hypervolume [13] are also known to be effective. More recently, research into a broader array of optimisation techniques such as estimation of distribution algorithms (EDAs), simulated annealing and ant colony optimisation has suggested that these alternative approaches may have distinct advantages on some problems [9].

In this article, we introduce, describe and benchmark a new algorithm for MOO problems called MOPC/D. It is an enhancement of the multiobjective probability collectives (MOPC) framework proposed in [1], with the key development being the use of decomposition as the primary method for trading-off the objectives. As a Probability Collectives [10] based algorithm, the fundamental objects in MOPC/D are probability distributions rather than solution vectors. The goal of an MOPC/D optimisation is to manipulate, through distribution sampling and local search techniques, a population of initially broad parametric distributions such that they become highly peaked around regions of Ω_1 in the neighbourhood of the Pareto set. This is in contrast to typical MOEAs and most existing probabilistic approaches such as EDAs, which instead seek to construct a high quality approximation to the Pareto set using a population of points. MOPC/D is oriented towards the common scenario in industry where the objective functions are the overwhelmingly significant factor in computational expense. For cheaper fitness functions, the lower computational overhead of a point-based approach may be preferable to the fidelity afforded by a probabilistic model.

We show how information contained within the probability distributions can be used to inform detailed searches of highly localised regions of the decision space. We introduce a new approach to regularisation and the fitting of parametric distributions that balances the simultaneous requirements of convergence and diversity, whilst also allowing the search to be conducted to a specified resolution over each decision variable. It is shown that the standard PC formulation, which is known to be effective on lower dimensional problems, can be recov-

978-1-4673-5889-7/13/$31.00

ered as a limiting case of the MOPC/D parameters. This means that existing regularisation techniques that are implemented using a cooling schedule (as in MOPC) or cross-validation (as outlined in [3]) are fully consistent with MOPC/D. In addition to using a mixture model representation for increased flexibility in addressing disparate fitness landscapes, we show how a traditional gradient based local search strategy can form an integral part of MOPC/D without prejudicing search diversity.

II. Probability Collectives

The aim of a Probability Collectives (PC) optimisation is to choose the set of parameters θ, defining the parametric distribution $q_\theta(\mathbf{x})$, such that $q_\theta(\mathbf{x})$ is the best possible approximation to the fitness landscape $p_\beta(\mathbf{x})$ where β is a regularisation parameter. A standard measure of disparity between two distributions is the Kullback-Leibler divergence given by (2). As $p_\beta(\mathbf{x})$ is independent of θ, this optimisation can be stated as the cross-entropy minimisation problem (3).

$$KL(p_\beta \parallel q_\theta) = \int_\Omega p_\beta(\mathbf{x}) \ln\left(\frac{p_\beta(\mathbf{x})}{q_\theta(\mathbf{x})}\right) d\mathbf{x} \tag{2}$$

$$\arg\min_\theta - \int_\Omega p_\beta(\mathbf{x}) \ln\left(q_\theta(\mathbf{x})\right) d\mathbf{x} \tag{3}$$

Formally, q_θ can come from the set of all possible parametric models although a smaller subset is usually chosen; here we consider q_θ as belonging to the set of all possible multivariate Gaussian mixture models. For the case where only a single mixture component is present, the integral can be approximated using importance sampling as a sum over a set of data points (4), where the weightings w_i are inversely proportional to the density of the proposal distribution $h(\mathbf{x})$. However, MOPC/D uses a variety of local search operators and therefore does not perform pure importance sampling. Instead, we assume that the proposal distribution is large in comparison with the sampling region and can therefore be taken to be uniform (5). Making this approximation has the additional benefit of avoiding the known issues in importance sampling relating to the distribution tails [11], which can be significant in high dimensional spaces where the sampling density is low; typically the case in a multiobjective problem.

$$\arg\min_\theta - \sum_i w_i \ln\left(q_\theta(\mathbf{x}_i)\right) \tag{4}$$

$$w_i = \frac{p_\beta(\mathbf{x}_i)}{h(\mathbf{x}_i)} \rightarrow p_\beta(\mathbf{x}_i) \tag{5}$$

If several mixture components are being fitted by expectation maximisation, it is necessary to introduce a latent variable z_i that dictates which component the sample is drawn from. This optimisation problem is specified by (6), where the distribution $Q_i(z_i)$ is over all the possible values of z_i. As described by [3] Jensen's inequality permits Q_i to be taken outside the logarithm, forming a lower bound which can be made tight by choosing $Q_i(z_i)$ to be the distribution $p(z_i \mid \mathbf{x}_i)$.

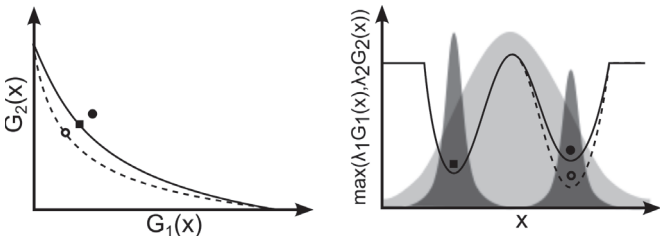

Fig. 1. In noisy problems, the Pareto set may be sensitive to small variations in the Pareto front. A small perturbation (dotted) to the fitness landscape around part of the Pareto front can cause well-separated points to exchange places in the Pareto set. The benefits of using a probabilistic mixture model rather than a single point are evident in this scenario.

$$\arg\min_\theta - \sum_i w_i \ln \sum_{z_i} Q_i(z_i) \frac{q_\theta(\mathbf{x}_i)}{Q_i(z_i)} d\mathbf{x} \tag{6}$$

There are several advantages to pursuing a distributional, rather than point-based approach within the decomposition framework for multiobjective problems. Firstly, it is possible at any stage in the optimisation to estimate the likelihood that an arbitrarily chosen region of Ω_1 contains a solution whose objective vectors are close to any arbitrarily specified subsection of the Pareto front. Similarly, it is possible to estimate correlations between the decision variables at arbitrarily chosen points on the Pareto front directly from the covariance matrices $\{\mathbf{\Sigma}\}$. The extent of each distribution component, $\det(\mathbf{\Sigma})$, allows a confidence bound for the current approximation to be determined. In contrast, a typical point-based approach reveals little about the general structure of the decision space with respect to the Pareto front, or the accuracy of the solution, other than providing the current 'best' single solution for each scalarised subproblem.

Some multiobjective problems are so complex that a full characterisation of the Pareto set may be unachievable; in such cases a knowledge of covariances amongst the decision variables is likely to be a significant improvement on what would otherwise be available. In other problems (e.g. those involving noise), the flexibility afforded by a distributional approach may be particularly effective at accelerating convergence. For example in the bi-objective/single decision variable problem of Fig. 1, the distribution $p_\beta^\lambda(\mathbf{x})$ on the RHS approximates the structure of the regularised map for the subproblem associated with the aggregation vector λ, which relates to a particular point on the Pareto front (LHS). In this type of problem, where disparate regions of the decision space perform similarly with respect to the same fitness function, a multi-modal distribution promotes a simultaneous search over all the relevant regions of the space, whereas the point based approach would typically conduct a search predominantly in one of the optima.

III. MOPC/D Algorithm

The fundamental objects in MOPC/D, which is described by Algorithm 1, are given in Table 1. As an MOO algorithm based on decomposition, MOPC/D uses a set of well spaced vectors $\{\lambda\}^k$, $k \in \{1, ..., m\}$ chosen from the set of all positive vectors $\{\Lambda\}$ on the b-dimensional unit simplex that

Object	Description
$\{\theta\}^k$	Population of distribution parameters
$\{\mathbf{x}', \mathbf{y}'\}_i^k$	Active points
$\{\mathbf{x}, \mathbf{y}, l\}_i^k$	Inactive points and most recent usage
$\{\lambda\}^k$	Aggregation vector
$\{N\}^k$	Vector of near neighbours
D_k	Sampling Density

TABLE I
DATA STRUCTURES AND PARAMETERS USED IN MOPC/D.

characterise specific directions in the objective space. In addition to vectors representing the b single objective problems, well-spaced vectors representing convex combinations of the objectives are iteratively added to the set $\{\lambda\}$ using Algorithm 2 until $|\{\lambda\}| = m$. Associated with each weight vector are the set of parameters $\{\theta\}^k$ that define the probability distribution q_θ^k. MOPC/D uses multivariate Gaussian mixture models, so the required parameters are $\theta \equiv \{\phi, \mu, \mathbf{\Sigma}\}_j$ where ϕ are the mixture weights, μ are the distribution means and $\mathbf{\Sigma}$ are the covariance matrices of each j-indexed mixture component.

In game theoretic terminology, each distribution can be viewed as an agent with its own individual task defined by its weight vector. In systems where the relationship between tasks are known, it is often beneficial for agents to exchange information with other agents working towards closely allied goals. The success of MOEA/D demonstrates that defining neighbourhood relationships between aggregation vectors can yield good performance on difficult MOO problems. However, instead of performing crossover operations amongst neighbouring solution points, the analogue in MOPC/D is that some samples associated with a particular distribution q_θ^k are made available to its neighbouring distributions $q_\theta^{\{N\}^k}$ during the update process. This reflects the likelihood that high fitness samples with respect to an aggregation vector will typically also be of high fitness with respect to its neighbouring subproblems; utilising neighbourhood relationships in this way can significantly increase the effective sampling density. Having determined the set of aggregation vectors $\{\lambda\}$, Algorithm 2 iteratively deduces the indices of the m_{nb} most closely related subproblems for each λ and assigns these to the neighbourhood set $\{N\}^k$. Similarity amongst subproblems is defined by Euclidean distance between weight vectors.

Also associated with each λ is a spacing parameter D that characterises how well spaced each member of the population is from its neighbours in objective space, allowing sampling to be actively directed towards less well resolved subproblems.

Algorithm 1 MOPC/D

procedure MOPC/D
  INITIALISE WEIGHTS(m,m_{nb})
  for $k \leq m$ **do**
    $\{\mathbf{x}'\}^k \leftarrow Rand\,(100) \in \Omega_1$
    $\{\mathbf{y}'\}^k = \mathbf{G}\,(\mathbf{x})$
  end for
  while $Evaluations < MaxEvaluations$ **do**
    for $k \leq m$ **do**
      $D_k = \langle \{\mathbf{y}'\}^k - \{\mathbf{y}'\}^{\{N\}^k} \rangle$
    end for
    $D \leftarrow D / \sum_k D_k$
    for $k \leq m$ **do**
      SELECT($\{\mathbf{x}', \mathbf{y}'\}$, $\{\mathbf{x}, \mathbf{y}, l\}$,$k$,$\lambda$,$N$)
      FIT($\{\mathbf{x}', \mathbf{y}'\}$,$\beta_{ratio}$,$k$,$N$)
      SAMPLE(q_θ,$\{\mathbf{x}', \mathbf{y}'\}$,$k$,$N$,$D$)
    end for
  end while
end procedure

Algorithm 2 Weight and Neighbourhood Assignment

procedure INITIALISE WEIGHTS(m,m_{nb})
  $\{\lambda\} = \emptyset\,, \{N\} = \emptyset$
  for all $\|\Lambda\|_\infty = 1$ **do**
    $\{\lambda\} \leftarrow \{\lambda\} \cup \Lambda$
  end for
  while $|\{\lambda\}| < m$ **do**
    $\{\lambda\} \leftarrow \{\lambda\} \cup \underset{\lambda^{new} \in \{\Lambda\}}{\arg\max} \left(\min_{\nu \in \{\lambda\}} \|\lambda^{new} - \nu\| \right)$
    $\forall\, \Lambda \mid \|\Lambda\|_1 = 1$
  end while
  for all k **do**
    while $\left|\{N\}^k\right| < m_{nb}$ **do**
      $\{N\}^k \leftarrow \{N\}^k \cup \underset{j\,:\,j \notin k \cup \{N\}^k}{\arg} \left(\underset{\nu^j \in \{\lambda\}}{\arg\min} \left\|\lambda^k - \nu^j\right\| \right)$
    end while
  end for
end procedure

A. Selection

Probability Collectives algorithms operate by sampling from a distribution and adding the evaluated decision/objective vector tuples to the distribution generating set. When updating the distribution, all previous samples would typically be used to form a best fit given all of the available information, which is desirable from a Bayesian viewpoint. However, on very large problems, storing and fitting to all previously evaluated samples can become infeasible. Instead, a maximum archive size is typically defined and solutions are removed when this is exceeded, which can be counter-productive because it does not guarantee retention of higher fitness samples. If the fitness landscape is sufficiently smooth the optimisation may recover due to other high fitness samples in the general neighbourhood. Unfortunately, on most multiobjective problems the features of the fitness landscape are poorly resolved by the available sampling density, increasing the likelihood of the distribution becoming trapped in a local optimum.

In contrast, EDAs generally behave more like an MOEA in that very few of the evaluated samples points are stored and utilised. There is therefore no need for a regularisation parameter to drive the distribution towards higher fitness regions,

because the 'forgotten' lower fitness samples effectively attract a permanent zero weighting. This can cause problems with search stagnation because there is no in-built mechanism for diversity maintenance, so concepts such as ε-dominance in the objective space are sometimes invoked to prevent premature convergence around a local optimum. With far fewer stored samples, it is more difficult to accurately extract correlation information amongst the decision variables.

MOPC/D incorporates aspects of both approaches. Consistent with the PC approach, as many samples as possible are stored, but only a fraction are used at any one time for distribution fitting. For each subproblem, the evaluated points are split into two disjoint sets: the first, $\{(\mathbf{x}', \mathbf{y}')\}^k$, is a tuple of n' decision/objective space vector pairs that are actively involved in updating the distribution q_θ^k, whilst the second, $\{(\mathbf{x}, \mathbf{y})\}^k$, is a set of inactive vectors that carry a zero weighting in the fitting process. The membership of each set is determined using Algorithm 3 and Algorithm 4.

Algorithm 3 Select High Fitness, Well Spaced Points

procedure SELECT($\{\mathbf{x}', \mathbf{y}'\}$, $\{\mathbf{x}, \mathbf{y}, l\}$,$k$,$\lambda$,$N$)
  $\{\mathbf{X}, \mathbf{Y}\} \leftarrow \{\mathbf{x}', \mathbf{y}'\}^k \cup \{\mathbf{x}, \mathbf{y}\}^{k \cup \{N\}^k}$
  $order \leftarrow sort\left[f\left(\mathbf{Y}, \lambda^k\right)\right]$
  $\{\mathbf{X}, \mathbf{Y}\} \leftarrow \{\mathbf{X}\left(order\right), \mathbf{Y}\left(order\right)\}$
  $\{\mathbf{x}', \mathbf{y}'\}^k \leftarrow$ DISTANCE SELECT($\{\mathbf{X}, \mathbf{Y}\}, n'$,$\varepsilon$)
  $\{\mathbf{x}, \mathbf{y}, l\}^{k \cup \{N\}^k} = \{\mathbf{x}, \mathbf{y}, l+1\}^{k \cup \{N\}^k} \setminus \{\mathbf{x}', \mathbf{y}'\}^k$
end procedure

Candidates for the active set are iteratively chosen from the union of the existing active set and the inactive sets across $k \cup \{N\}^k$, according to both their Chebychev fitness with respect to λ^k and their spacing from previously selected candidates. The relative merits and failings of different scalarised fitness functions are well understood; the Chebychev function (7) is used for selection in MOPC/D because it can treat non-convex problems (unlike linearly weighted combinations) and does not need any additional parameters (such as penalty functions in boundary intersection methods). The only parameter which must be learnt is the utopia (or ideal) point y_j^* for each objective, which in general is an unknown. It is essential that aggregation based algorithms learn this quickly because it can significantly affect the placement of search effort.

$$f\left(\mathbf{y}, \lambda\right) \quad = \quad \max_j \left(\lambda\left(j\right) \left| y\left(j\right) - y^*\left(j\right)\right|\right) \tag{7}$$

Algorithm 3 describes how points in the active set are chosen to be of high fitness and broadly spaced. Having ranked the candidate points according to fitness against (7) for λ^k, we introduce the resolution parameter ε in decision space that defines a minimum separation, in every dimension, between any two selected points in the active set. At each iteration, the best fitting point that meets the minimal spacing requirement is selected. By applying the ε criterion in the decision space, it is possible customise the search over the variables to provide the desired resolution with respect to each decision variable. Choosing a coarser resolution simplifies the problem and promotes diversity, but may result in the solution being insufficiently localised and a slow rate of convergence. This method guarantees that after marginalising q_θ^k over any number of dimensions, the resulting distribution is fitted to points with a guaranteed minimum range of $\varepsilon n'$. In the limit where $\varepsilon \to 0$ and $n' \to \infty$ the standard Probability Collectives formulation is recovered. Similarly, in the limit where $n' = 1$ the resolution parameter ε is not relevant, and something more akin to a typical EDA/MOEA selection routine is specified.

By dividing the available points into two sets broadly representing 'successful' and 'unsuccessful' candidates, there is an opportunity to conduct any necessary culling of the stored solutions in an informed way. In general, it is preferable to retain solutions that are of high fitness and well spaced rather than the most recent solutions. To do this, the additional integer variable l is stored for each tuple in the inactive set to count the number of iterations over the range $k \cup \{N\}^k$ since that tuple was last successful in gaining a position in an active set; therefore $\{\mathbf{x}, \mathbf{y}, l\}^k$ is stored rather than $\{\mathbf{x}, \mathbf{y}\}^k$. When $l > n_{decay}$, where n_{decay} is a constant, that tuple is deleted. This is not a parameter that needs tuning; rather it should be as large as possible, subject to computational resources, because initially excluded points may become important at a later stage due to their large spacing from higher fitness points.

Algorithm 4 Minimum Spacing Criterion

procedure DISTANCE SELECT($\{\mathbf{X}, \mathbf{Y}\}, n'$,$\varepsilon$)
  $\{\mathbf{x}', \mathbf{y}'\}^k = \emptyset$, $i = 1$, $r = 1$
  while $i < n'$ **do**
    $\varepsilon^* = \varepsilon$
    while $r \leq |\{\mathbf{X}, \mathbf{Y}\}|$ **do**
      if $\min_j \left(\min_q \left| \mathbf{X}_r\left(q\right) - \mathbf{x}'_j(q) \right| \right) < \varepsilon^*$ **then**
        $r \leftarrow r + 1$
      else if $r = |\{\mathbf{X}, \mathbf{Y}\}|$ **then**
        $\varepsilon^* \leftarrow 0.9\varepsilon$
        $r \leftarrow 1$
      else
        $\{\mathbf{x}', \mathbf{y}'\}_i^k = \{\mathbf{X}, \mathbf{Y}\}_r$
      end if
    end while
    $i \leftarrow i + 1$
  end while
end procedure

B. Fitting

We now address the key part of MOPC/D, that of updating the distribution parameters $\theta_j^k \equiv \{\phi, \mu, \mathbf{\Sigma}\}_j$. A total of $n'\left(m_{nb}+1\right)$ tuples are used to derive the best fit parameters for each distribution q_θ^k. These include all of the active tuples associated with the subproblem λ^k, but also those associated with the neighbouring subproblems $\lambda^{\{N\}^k}$. Together, these form the set of tuples $\{\mathbf{X}, \mathbf{Y}\}$. The influence of the neighbourhood size m_{nb} on the fitting process is evident: smaller neighbourhoods are likely to generate distributions which are

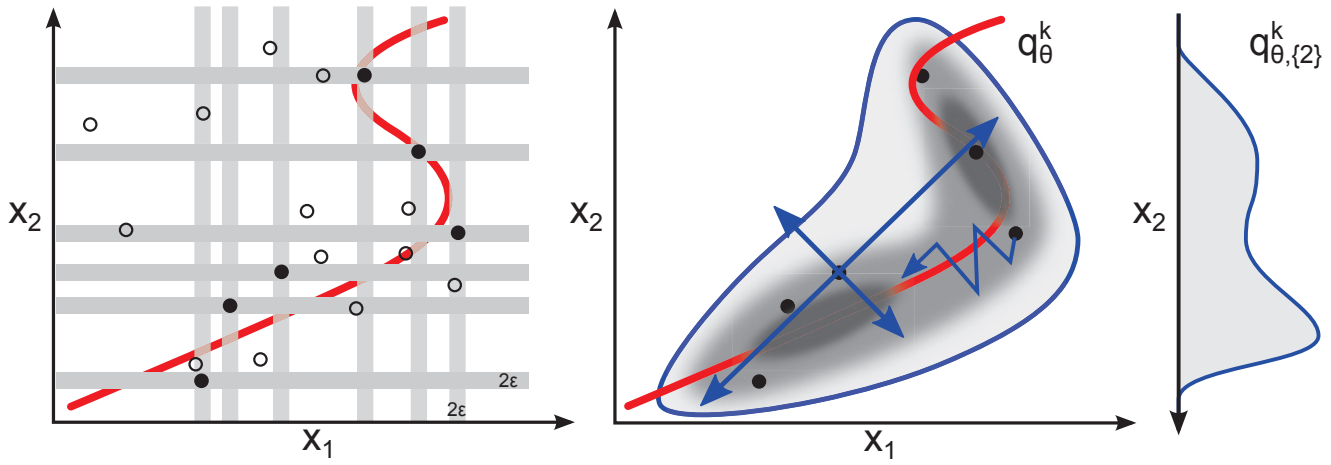

Fig. 2. MOPC/D aims to fit a Gaussian mixture model to a region of the Pareto set (red) using points associated with the aggregation vectors $\{\lambda\}^{k\cup\{N\}^k}$. Points are selected (left) according to their Chebychev fitness and subject to a minimum inter-point spacing of ε in each dimension within the set. A new distribution q_θ is fitted to these points (right), which is then searched using one of four different methods (blue): direct sampling, sampling along an eigenvector, sampling from a marginal distribution and traditional local optimisation.

better localised around a specific subproblem, whereas larger neighbourhoods use more data points and are likely to generate better quality and more representative distributions.

The fitted Gaussian mixture model is a combination of two models that carry equal weight. The first is an analytically fitted single component model, whilst the second is the multi-component model fitted by Expectation Maximisation which returns the smallest Bayesian Information Criterion (BIC). Whilst the multi-component model will typically be a better fit to the data, there is still a propensity to overfit because the number of data points n' being used to generate the model is usually substantially fewer than the number of free parameters in the covariance matrix. For this reason, it is also necessary to add a small regularisation to each covariance matrix diagonal.

Algorithm 5 Distribution Update

procedure FIT($\{\mathbf{x}',\mathbf{y}'\}$,$\beta_{ratio}$,$k$,$N$)
  $\{\mathbf{X},\mathbf{Y}\} \leftarrow \{\mathbf{x}',\mathbf{y}'\}^{k\cup\{N\}^k}$
  $H \leftarrow h(\{\mathbf{Y}\})$
  $\beta^k = \min_\beta \left| \frac{\min[\exp(-\beta H)]}{\max[\exp(-\beta H)]} - \beta_{ratio} \right|$
  $w = \exp\left(-\beta^k H\right) / \sum_{\{H\}} \exp\left(-\beta^k H\right)$
  $\{\phi,\mu,\mathbf{\Sigma}\}_1^k \leftarrow Gaussian\ Fit\{\mathbf{X},\mathbf{Y},w\}$
  $\{\phi,\mu,\mathbf{\Sigma}\}_{[2,:]}^k \leftarrow EM\ Fit\ +\ BIC\{\mathbf{X},\mathbf{Y},w\}$
  $\{\theta\}^k \leftarrow \{\phi,\mu,\mathbf{\Sigma}\}^k$
end procedure

Each subproblem has an optimal point $\mathbf{x}^*$ with respect to (7) for λ^k somewhere within Ω_1. A significant challenge in Probability Collectives optimsation is determining the rate at which the distribution collapses around $\mathbf{x}^*$. This is controlled by the regularisation parameter, or inverse distribution temperature β, which determines the weight w_i that each point $\mathbf{X}_i$ carries in the fitting process according to (8). Typically, its value would increase as the optimsation proceeds according to a multiplicative update rule in a simulated annealing type approach, or be adjusted dynamically using a supervised learning technique such as cross-validation. In large search spaces it can be difficult for either method to accurately determine the most appropriate value for β, because there are often insufficiently many samples available to resolve the important features of the fitness landscape. This is a significant problem in the existing multiobjective implementation of Probability Collectives, because the convergence towards $\mathbf{x}^*$ is driven entirely by this parameter. Worse, if β becomes too large, a point that is difficult to define, then $\mathbf{\Sigma}$ can become highly ill-conditioned and cause the algorithm to fail.

$$w = \exp\left(-\beta h\left(\{\mathbf{Y}\}\right)\right) \tag{8}$$

$$h\left(\{\mathbf{Y}\}\right)(i) = \max_{b\neq i, \mathbf{Y}_b \not\succ \{\mathbf{Y}\}} \left(\min_j \left(Y_i(j) - Y_b(j) \right) \right) \tag{9}$$

MOPC/D is far less sensitive to the choice of β because the points involved in the fit are already known to be well spaced and of high fitness, so setting $\beta = 0$ and not using a weighting function would be a perfectly valid way of undertaking the optimsation. However, this is an opportunity to address the known issue in multiobjective optimisation of different fitness functions tending to vary in effectiveness across different problems. Whilst the scalarised Chebychev function is used for point selection, we instead use the modified maxmin function h for distribution weighting [12]; a dominance based fitness function that returns a numerical value and is known to perform well on some multiobjective problems. It is particularly suitable for biasing a fit amongst a small set of points as it describes relative rather than absolute performance. Instead of supplying a fixed parameter value for β, it is calculated dynamically for each subproblem (β^k) such that the weight ratio between the worst and best fitting points w_i^{worst} and w_i^{best} in $\{\mathbf{X}\}$ is fixed to be the constant β_{ratio}. The resulting distribution is therefore guaranteed to assign a reasonably large weighting to every selected point.

C. Sampling

The updated population of distributions is now used to generate a new set of sample points. Unlike an MOEA, there is additional information in the stored distributions of MOPC/D that can be used to inform the generation of new samples, for which there are three 'operators' in MOPC/D in addition to a standard local search algorithm. The first involves sampling directly from q_θ^k; the only sampling method typically used in PC optimisation. Its primary advantage is the ability to generate well spaced samples based on the best currently available information, but it is also relatively resistant to becoming trapped in a local optimum. Unfortunately, in multiobjective problems with large sample spaces its convergence tends to be relatively slow when compared with MOEA search operators unless the distribution is forcibly narrowed through regularisation, negating its principal advantage. In MOPC/D this is not necessary because two further more localised search operators are available.

$$\mathbf{Z}_x^{marg} = \begin{cases} \sim q_{\theta,\{m\}}^k & \forall\, m \in \{M\} \\ \mathbf{Z}_x & \text{Otherwise} \end{cases} \tag{10}$$

$$q_{\theta,\{m\}}^k = \int_{\Omega_1} q_\theta^k\left(\mathbf{x}'\right) dx'_{m \notin \{M\}} \tag{11}$$

The second operator involves reducing the dimensionality of the problem by marginalising the distribution over some of the decision variables (10), leaving these elements fixed to those of an existing non-dominated sample $\mathbf{x}_{seed}\,(m \notin M)$ in the neighbourhood. For this study, the set of active dimensions $\{M\}$ was chosen to be a single randomly chosen dimension at each iteration; it would be relatively straightforward to implement supervised learning to select the dimensions which are most in need of refinement. The choice of a Gaussian parametric form greatly simplifies the marginalisation procedure.

Algorithm 6 Generate New Samples

procedure SAMPLE(q_θ,$\{\mathbf{x}', \mathbf{y}'\}$,$k$,$N$,$D$)
  $\{\mathbf{Z}_x\} \leftarrow [nD_k\alpha\,(1)]$ samples from q_θ^k
  $\mathbf{x}_{seed} \leftarrow \{\mathbf{x}'\}^{k\cup\{N\}^k}\ \forall\ h\left(\{\mathbf{y}'\}^{k\cup\{N\}^k}\right) \leq 0$
  $\{\mathbf{Z}_x\} \leftarrow \{\mathbf{Z}_x\} \cup [nD_k\alpha\,(2)]$ samples from $\mathbf{Z}_x^{marg}$
  $\{\mathbf{Z}_x\} \leftarrow \{\mathbf{Z}_x\} \cup [nD_k\alpha\,(3)]$ samples along $eig\,(q_\theta)$
  $\{\mathbf{Z}_y\} = G\,(\{\mathbf{Z}_x\})$
  if $rand < \gamma$ **then**
    $\{\mathbf{Z}_x, \mathbf{Z}_y\} \leftarrow \{\mathbf{Z}_x, \mathbf{Z}_y\} \cup$ SQP local optimisation
  end if
  $\{\mathbf{x}, \mathbf{y}, l\}^k \leftarrow \{\mathbf{x}, \mathbf{y}, l\}^k \cup \{\mathbf{Z}_x, \mathbf{Z}_y, 0\}^k$
end procedure

The third operator also involves choosing one of the non-dominated samples within $\{\mathbf{x}'\}^{k\cup\{N\}^k}$ as a seed point. Approximations for correlations amongst the variables are extracted through a spectral decomposition of the (analytically fitted) first covariance matrix in the mixture model: $\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1} = \mathbf{\Sigma}_1^k$. One of the principle components $\mathbf{p}_i$ (columns of $\mathbf{Q}$) is selected at random with a probability $P \propto \sqrt{\Lambda_{ii}}$ according to the standard deviation along that component. Then, a new sample is drawn from a one dimensional normal distribution defined along the line of $\mathbf{p}_i$ at the seed point; its mean is the seed point and the standard deviation is again taken to be $\sqrt{\Lambda_{ii}}$. The relative frequencies of the direct, marginal and principle component searches are specified by the parameter vector α.

Finally, the use of purely local search techniques is known to improve many optimisation algorithms. In MOPC/D, local optimisation is applied at each iteration to a randomly chosen point in $\{\mathbf{x}'\}^k$ with a probability γ against the fitness metric $f\left(\mathbf{x}, \lambda^k\right)$. The initial guess is chosen randomly amongst all possible seed points, rather than known a nondominated point, to enable the algorithm to determine the depth of a variety optima in which the points in $\{\mathbf{x}'\}^k$ may reside.

IV. Results

MOPC/D was evaluated using the 2D and 3D unconstrained problems from the CEC2009 competition test suite [4]. Since MOPC/D gives an information-rich probabilistic approximation for the Pareto set, standard point based metrics are not the ideal performance measure. However, we seek to compare performance with state-of-the-art MOEAs, and hence use the 'IGD' measure specified for the CEC2009 competition. IGD quantifies the proximity of the approximation (in the final iteration) to the known Pareto front. We exploit comparative IGD results for these problems for 13 state of the art MOEAs evaluated in the CEC2009 competition, summarised in [5].

To save space we tabulate here MOPC/D results against only MOPC (the previous MOO algorithm in the PC framework), MOEA/D and GDE3, just two of those evaluated in [5], but we also quantify MOPC/D's overall performance in comparison to all 13 algorithms summarised therein.

Visualisations of q_θ^k for the bi-objective problems are also presented to give a qualitative indication of how well the set of distributions $\left\{q_\theta^k\right\}$ approximate the Pareto set. Whilst impossible to visualise a 30-dimensional distribution in 2 dimensions, most of the CEC problems are highly symmetric about the first dimension (UF1-UF7) so a meaningful representation can be achieved through a heat map of x_1 and one other dimension.

The tunable parameter values used in the optimisation are given in Table 2. In addition, the decay rate and sampling rate (the number of samples taken before the distribution is refitted) are set to $n_{decay} = 100$ and $n = 10$. Better performance may be possible with larger n_{decay} and smaller n respectively; these settings allowed for reasonable memory usage and run time on the test problems, arising from basic intuition and preliminary work with early and different versions of MOPC/D.

According to the IGD metric, MOPC/D performs significantly better than the original MOPC algorithm on all of the problems. The performance is comparable to the sampled state of the art MOEAs, being neither significantly better or worse when considered across all of the problems in the test suite. The latter remark is also the case when compared with all 13 algorithms evaluated in [4], [5]. An indicative quantification

Problem	MOPC/D	MOPC	MOEA/D	GDE3
UF1	0.0097 ± 0.0029	0.0242 ± 0.0036	0.00435 ± 0.00029	0.005342 ± 0.000342
UF2	0.0101 ± 0.0023	0.0386 ± 0.0015	0.00679 ± 0.00182	0.011953 ± 0.001541
UF3	0.0128 ± 0.0087	0.1740 ± 0.0193	0.00742 ± 0.00589	0.106395 ± 0.01290
UF4	0.0426 ± 0.0020	0.1151 ± 0.0061	0.06385 ± 0.00534	0.026506 ± 0.000372
UF5	0.1460 ± 0.0340	0.5017 ± 0.0294	0.18071 ± 0.06811	0.039281 ± 0.0003947
UF6	0.0724 ± 0.0193	0.1115 ± 0.0155	0.00587 ± 0.00171	0.250913 ± 0.019573
UF7	0.0113 ± 0.0011	0.0521 ± 0.0030	0.00444 ± 0.00117	0.025228 ± 0.008891
UF8	0.0771 ± 0.0060	0.3691 ± 0.0122	0.05840 ± 0.00321	0.248556 ± 0.035521
UF9	0.0787 ± 0.0230	0.1514 ± 0.0044	0.07896 ± 0.05316	0.082482 ± 0.022485
UF10	0.3412 ± 0.0854	0.4489 ± 0.0181	0.47415 ± 0.07306	0.433261 ± 0.012323

TABLE III
MEAN IGD VALUES AND STANDARD DEVIATIONS OVER THE UNCONSTRAINED 2&3 OBJECTIVE PROBLEMS IN THE CEC2009 COMPETITION

Parameter	Value	Description
m	50	Population size where $k \in \{1, ..., m\}$
m_{nb}	5	Neighbourhood size
α	$(0.1, 0.45, 0.45)$	Search type vector
γ	0.01	Local search probability
n'	12	Number of non-zero weighted vectors
β_{ratio}	0.05	Distribution regularisation parameter
ε	0.0075	Search Resolution

TABLE II
MOPC/D PARAMETER SETTINGS USED FOR THE CEC2009 PROBLEMS

of how MOPC/D compares with the current state of the art can be obtained by recalculating the ranks for problems UF1–UF10 from [5] with MOPC/D included. When the mean of those 10 ranks are computed, MOPC/D's overall position is 1st place (out of 14), with performance broadly similar to MTS, DMOEA/DD and MOEA/D. These four algorithms form a clear leading group (Fig. 3) with mean ranks of 4.2, 4.3, 4.3 and 4.4 respectively. MOPC/D has the smallest standard deviation of rank across the problem set, a measure that informs an indicative (rather than statistically significant) comparison of the relative robustness of each algorithm.

These results suggest that Probability Collectives based optimisation is a realistic alternative to state of the art MOEAs on multiobjective problems. On the easier problems UF1, UF2 and UF7, MOPC/D was able to find excellent approximations to the reference set that were similarly well specified to the MOEAs - it is likely that performance at this level is constrained by the resolution parameter ε. On the problems known to be more challenging (from the results of the original competition) UF3, UF4, UF5 and UF6, MOPC/D split the two tabulated high performing MOEAs by being better than MOEA/D on UF4 and UF5 whilst being better than GDE3 on UF3 and UF6.

Fig. 3 shows how MOPC/D is able to generate a broadly accurate probabilistic description of the Pareto set on UF1, UF2, UF3 and UF6. On UF4 and UF7, the correct shape is evident in some regions but it is not as highly peaked as would be desired. On UF5, the distribution fails to identify the correct shape and has not converged, although the IGD performance is still reasonable. This suggests that the fitness landscape might be irregular over relatively short length scales, which is indeed the case. These images highlight the benefit of using a mixture model representation and neighbourhood relationships; in general the observations are mirrored in the other dimensions that are not plotted.

V. CONCLUSION

MOPC/D, a new Probability Collectives algorithm based around decomposition, has been presented and shown to be competitive with leading MOEAs on a variety of test problems. By introducing decomposition, local, marginal and principal component searches, the algorithm is able to perform well across a broader array of problems than the original MOPC. The high performance of MOPC/D and flexibility of being able to use multiple search techniques within the one algorithm suggests that Probability Collectives inspired optimisation algorithms are strong candidates for MOO problems.

There are several ways in which it may be possible to improve performance. One would be to apply supervised learning to determine the optimal values for the sampling vector α and the spacing parameter ε for a particular problem so that pre-configuration is not required. Such an approach has proved highly beneficial in the MOEA context [6]. Another would be to use variational Bayesian methods, rather than cross-validation, for model selection in the distribution fitting process. Finally, it is likely that sampling efficiency could be improved by exploring more sophisticated methods for calibrating sampling density against subproblem complexity.

REFERENCES

[1] A.Waldock & D.Corne, *Multi-objective probability collectives*, Applications of Evolutionary Computation pp. 461-470, 2010

[2] Q.Zhang, W.Liu & H.Li, *The Performance of a New Version of MOEA/D on CEC09 Unconstrained MOP Test Instances*, IEEE CEC, 2009

[3] D.Wolpert, S.Bieniawski & D.Rajnarayan, *Probability Collectives in Optimization*, Santa Fe Institute Working Paper # 11-08-033, 2011

[4] Q.Zhang et al, *Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition*, Working Report CEC-887, University of Essex, 2009

[5] Q.Zhang & P.Suganthan, *Final Report on CEC'09 MOEA Competition*, http://dces.essex.ac.uk/staff/qzhang/MOEAcompetition/cecmoeafinalreport.pdf

[6] D.Hadka & P.Reed, *Borg: An Auto-Adaptive Many-Objective Evolutionary Computing Framework, Evolutionary Computation*, 2012

[7] L.Tseng & C.Chen, *Multiple Trajectory Search for Unconstrained/Constrained Multi-Objective Optimization*, IEEE CEC, 2009

[8] K.Deb, *A fast and elitist multiobjective genetic algorithm: NSGA II*, IEEE Transactions on Evolutionary Computation, 2002

[9] A.Zhou et. al., *Multiobjective evolutionary algorithms: A survey of the state of the art*, Swarm and Evolutionary Computation 1 pp. 32-49, 2011

[10] C.Huang et al., *A comparative study of probability collectives based multi-agent systems and genetic algorithms*, GECCO Proceedings of Conference on Genetic and Evolutionary Computation, 2005

[11] C.Robert & G.Casella, *Monte Carlo Statistical Methods 2nd Edition*, Springer Texts in Statistics, 2004

[12] R.Balling, *The Maximin Fitness Function; Multi-objective City and Regional Planning*, Lecture Notes in Computer Science, 2003

[13] E.Zitler & S.Künzli, *Indicator-Based Selection in Multiobjective Search*, Lecture Notes in Computer Science, 2004

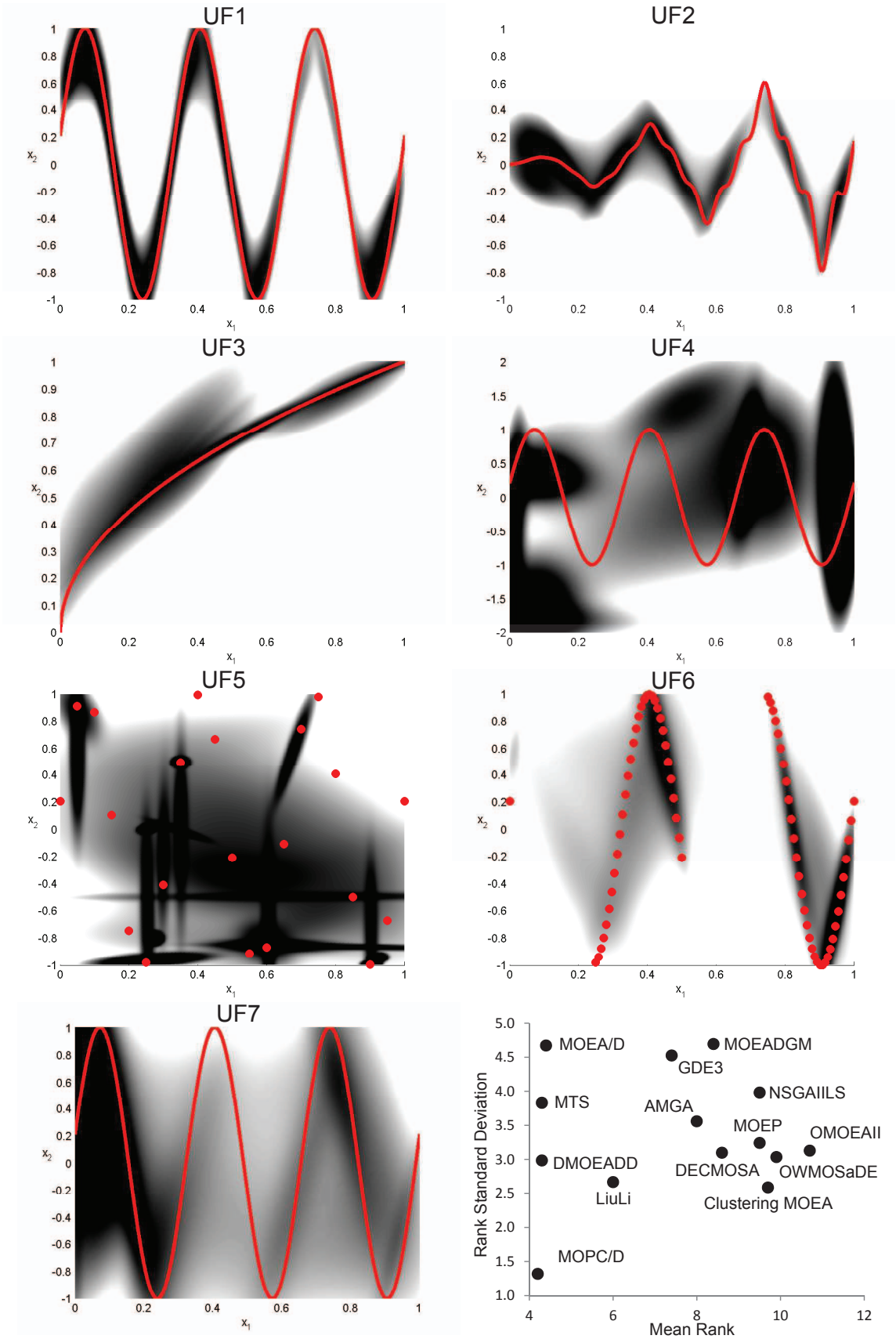

Fig. 3. The Pareto sets and log-likelihood distributions for problems UF1-UF7. The Pareto set is plotted in red and the MOPC/D approximation after 300,000 evaluations is plotted in greyscale (Top & Left). The relative mean rankings of MOPC/D and the MOEAs in the CEC2009 competition over UF1-UF10 and their rank standard deviations (Bottom Right).