Pareto Repeated Weighted Boosting Search for Multiple-Objective Optimisation

Scott F. Page, Sheng Chen, *Fellow, IEEE*, Chris J. Harris, and Neil M. White, *Senior Member, IEEE*

***Abstract*— A guided stochastic search algorithm, known as the repeated weighted boosting search (RWBS), offers an effective means for solving the difficult single-objective optimisation problems with non-smooth and/or multi-modal cost functions. Compared with other global optimisation solvers, such as the genetic algorithms (GAs) and adaptive simulated annealing, RWBS is easier to implement, has fewer algorithmic parameters to tune and has been shown to provide similar levels of performance on many benchmark problems. This contribution develops a novel Pareto RWBS (PRWBS) algorithm for multiple objective optimisation applications. The performance of the proposed PRWBS algorithm is compared with the well-known non-dominated sorting GA (NSGA-II) for multiple objective optimisation on a range of benchmark problems, and the results obtained demonstrate that the proposed PRWBS algorithm offers a competitive performance whilst retaining the benefits of the original RWBS algorithm.**

I. Introduction

In the work [1], a guided stochastic search or meta-heuristic algorithm, referred to as the repeated weighted boosting search (RWBS), was proposed to solve the complex optimisation problems with non-smooth and/or multi-modal cost functions. The advantages of RWBS [1] include ease of implementation, very few number of tuning parameters, and capable of achieving the levels of performance comparable with many standard benchmark techniques, such as the genetic algorithms (GAs) [2], [3] and the adaptive simulated annealing [4], [5]. RWBS is essentially a multi-start search technique [6], where the local optimisation mechanism is based on an iterative, adaptive, weighted convex combination. In conjunction with a reflection operator, the convex combination generates new solutions in a manner similar to the simplex method. The adaptive weight update process is a modified boosting technique [7]. A number of applications have been reported using the RWBS, which cover the diverse fields of machine learning, chaotic system stabilisation, image and signal processing as well as wireless communication designs [1], [8]–[18].

Although the RWBS algorithm has proved to be a very useful optimisation tool for diverse applications, its original form proposed in [1] is restricted to single-objective optimisation problems. This contribution proposes a novel extension to the original RWBS for the use in multiple-objective optimisation problems where no objective preference structure is available. The resulting algorithm maintains a set of Pareto-optimal solutions for subsequent inspection by the designer, similar to the well-known non-dominated sorting genetic algorithm (NSGA-II) [19]. The performance of the resulting algorithm, referred to as the Pareto-RWBS (PRWBS) algorithm, is assessed using some well-known benchmark problems. In comparison with the state-of-the-arts NSGA-II algorithm, the proposed PRWBS is shown to offer a promising level of performance in solving these multiple-objective optimisation problems whilst retaining the attractive properties of the original RWBS version.

The generic multiple-objective problem considered in this contribution is described as follows:

$$\min_{\mathbf{u}\in\mathsf{U}} f(J_1(\mathbf{u}), J_2(\mathbf{u}), \cdots, J_N(\mathbf{u})) \tag{1}$$

where $\mathbf{u} = [u_1 \ u_2 \cdots u_n]^{\mathrm{T}} \in \mathbb{R}^n$ is the n-dimensional vector of bounded decision variables to be optimised, U denotes the feasible set of $\mathbf{u}$, $J_i(\mathbf{u})$ is the i-th objective function, N is the number of objective functions, and f is the objective preference function which may or may not be present. Evaluation of the objective functions may be analytic or procedural, and the cost functions are not necessarily continuous or differentiable.

If *a priori* information regarding the relative importance of the different optimisation objectives is available, the multiple-objective optimisation problem (1) can be reformulated as a single-objective one, using a simple weighting method. Techniques which operate in this manner can be termed 'non-Pareto methods' as they search for solutions to surrogate problems. However, if preference information is not available or the nature of the Pareto-frontier is of direct interest, then the optimisation algorithm must generate a *set* of Pareto-optimal solutions. Ideally, the solutions should be well distributed across the Pareto-frontier[1]. These methods can be termed 'Pareto methods'. The solution set can then be used to consider which solution is most appropriate for the particular problem and to implicitly infer some relative importance of the objectives. Several methods have been proposed to adapt common population based stochastic search techniques to generate Pareto-optimal sets [20].

S.F. Page was with School of Electronics and Computer Science, University of Southampton, when the work reported in this paper was carried out. He is now with Waterfall Solutions Ltd, Guildford, Surrey GU2 9JX, U.K. (E-mail: scottpage888@gmail.com).

S. Chen, C.J. Harris and N.M. White are with School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (E-mails: sqc@ecs.soton.ac.uk, cjh@ecs.soton.ac.uk, nmw@ecs.soton.ac.uk).

The Authors acknowledge the support of Phase 1 of the UK MOD DIF DTC under Project 8.1 *Active Sensor Management*.

[1]What is meant by 'well distributed' is often problem specific, but a reasonably uniform distribution over the Pareto-frontier may be deemed desirable.

There are two main aspects to designing an efficient algorithm for Pareto-optimisation. Firstly, the algorithm needs to embody a mechanism which drives solutions towards the Pareto-frontier and, secondly, there needs to be a mechanism which ensures a good distribution of solutions across the frontier. Typically, a form of Pareto-ranking or Pareto-sorting is used to guide the optimisation towards the frontier [20]. These techniques effectively modify the cost value or fitness value for a solution depending on its performance *relative* to other solutions in the set, in contrast to the absolute notion of optimality used in conventional optimisation. Solutions which are 'non-dominated' or mildly dominated (i.e. only dominated by a limited number of other solutions) are attributed a higher fitness or lower cost than those which are strongly dominated. This promotes the generation of more non-dominated solutions. Distribution of solutions across the Pareto-frontier is commonly achieved using 'sharing' or 'niche' methods [20], [21]. Sharing methods distribute an individual's fitness depending on how many solutions are nearby it[2], thus encouraging spread. A difficulty with these sharing techniques is that the user must define the so-called 'sharing parameter' [20]. In general, manual fixing of the sharing parameter requires knowledge of the objective function and adds to the tuning complexity of the optimisation algorithm. In contrast, a distance based measure is used in [19] which is completely parameterless.

II. Pareto RWBS for multiple objective problems

The detailed RWBS algorithm can be found in [1], which contains three algorithmic parameters: the population size P_{s}, the number of generations in the repeated search N_{g}, and the number of iterations in the weighted boosting search (WBS) N_{B}. As the RWBS is a population-based stochastic search method, it can be readily adapted to the Pareto-optimisation case by a number of modifications. These include the addition of a Pareto-ranking process and a mechanism which encourages distribution as well as the modified elitism process that retains a larger set of solutions between generations, instead of the single point in the original algorithm.

Specifically, all the population members are ranked relatively, in terms of Pareto-dominance, according to the 'fast-non-dominated-sort' procedure proposed in [19]. To encourage a good spread across the Pareto-frontier, the resulting Pareto-ranking of the i-th population member, R_i, and the mean distance from all the other points, D_i, as well as a scaling parameter, P_{r}, are used to compute a distance and ranking adjusted cost for the ith population member according to

$$\hat{J}_i = \frac{P_{\mathrm{r}} R_i}{D_i}, \ 1 \le i \le P_{\mathrm{s}}. \tag{2}$$

In the original RWBS algorithm for single-objective optimisation, the elitism process only retains the single best solution to the next generation. This elitism process must be modified so that a larger set of solutions is retained. In other words, in order to identify a suitable set of Pareto-optimal solutions, a record of potential solutions must be retained during each generation. To achieve this, the elitism process is extended so that a larger proportion of the current population is kept between each generation. This introduces another new parameter, P_{e}, known as the 'elitism count', which specifies how many population members are kept between generations.

The proposed PRWBS algorithm is constructed as follows. Specify the following algorithmic parameters: the population size P_{s}, the number of generations in the repeated search N_{g}, the number of iterations in the WBS N_{B}, the Pareto-ranking scaling P_{r}, and the elitism count P_{e}.

○ **Outer Loop : generations** for $g = 1 : N_{\mathrm{g}}$

- *Pareto generation initialisation*: Initialise the population by setting $\mathbf{u}_i^{(g)} = \mathbf{u}_{\mathrm{best},i}^{(g-1)}$ for $1 \le i \le P_{\mathrm{e}}$, and randomly generating the rest of the population members $\mathbf{u}_i^{(g)}$ for $P_{\mathrm{e}} + 1 \le i \le P_{\mathrm{s}}$, where $\big\{\mathbf{u}_{\mathrm{best},i}^{(g-1)}\big\}_{i=1}^{P_{\mathrm{e}}}$ denotes the set of the 'best' P_{e} solutions found in the previous generation. If $g = 1$, $\mathbf{u}_i^{(g)}$, $1 \le i \le P_{\mathrm{e}}$, are also randomly chosen.
- *Weighted boosting search initialisation*: Assign the initial weights $\delta_i(0) = \frac{1}{P_{\mathrm{s}}}$, $1 \le i \le P_s$, for the population. Calculate the cost function values for each point of the population set and for each objective function

$$J_{i,o} = J_o\big(\mathbf{u}_i^{(g)}\big), \ 1 \le o \le N, 1 \le i \le P_{\mathrm{s}}. \tag{3}$$

- **Inner Loop : weighted boosting search** for $t = 1 : N_{\mathrm{B}}$
  - *Step 1*: Pareto Boosting
    1) Perform **Pareto Ranking, Distance Measure and Cost Mapping** for the current population $\big\{\mathbf{u}_i^{(g)}, J_{i,o}, 1 \le o \le N\big\}_{i=1}^{P_{\mathrm{s}}}$. Specifically,
       a) Calculate the Pareto-ranking for each member of the population:

$$\{R_i\}_{i=1}^{P_{\mathrm{s}}} = \textbf{FastNonDominatedSort}\big\{J_{i,o}, 1 \le i \le P_{\mathrm{s}}, 1 \le o \le N\big\}, \tag{4}$$

       using the method proposed in [19].

       b) For each member of the population, compute the mean Euclidean distance to all the other points in the decision variable space:

$$D_i = \frac{1}{P_{\mathrm{s}}} \sum_{j \ne i} \|\mathbf{u}_i^{(g)} - \mathbf{u}_j^{(g)}\|, 1 \le i \le P_{\mathrm{s}}. \tag{5}$$

       c) Compute the distance and ranking adjusted costs $\hat{J}_i$, $1 \le i \le P_{\mathrm{s}}$, according to (2).
    2) Find $i_{\mathrm{best}} = \arg \min_{1 \le i \le P_{\mathrm{s}}} \hat{J}_i$, and denote $\mathbf{u}_{\mathrm{best}}^{(g)} = \mathbf{u}_{i_{\mathrm{best}}}^{(g)}$.
    3) Normalise the distance and ranking adjusted cost values:

$$\bar{J}_i = \frac{\hat{J}_i}{\sum_{j=1}^{P_{\mathrm{s}}} \hat{J}_j}, 1 \le i \le P_{\mathrm{s}}.$$

[2]Sharing can take place either in the decision space or the fitness space, although in some cases decision space sharing is preferable [22].

4) Compute a weighting factor $\beta(t)$ according to

$$\eta(t) = \sum_{i=1}^{P_s} \delta_i(t-1)\bar{J}_i, \quad \beta(t) = \frac{\eta(t)}{1-\eta(t)}.$$

5) Update the weights for $1 \le i \le P_s$

$$\delta_i(t) = \begin{cases} \delta_i(t-1)\beta(t)^{\bar{J}_i} & \text{for } \beta(t) \le 1, \\ \delta_i(t-1)\beta(t)^{1-\bar{J}_i} & \text{for } \beta(t) \le 1, \end{cases}$$

and normalise them:

$$\delta_i(t) = \frac{\delta_i(t)}{\sum\limits_{j=1}^{P_s} \delta_j(t)}, \quad 1 \le i \le P_s.$$

- *Step 2*: Pareto Parameter Update

1) Construct the (P_s+1)-th point using the formula

$$\mathbf{u}_{P_s+1} = \sum_{i=1}^{P_s} \delta_i(t)\mathbf{u}_i^{(g)}.$$

2) Construct the (P_s+2)-th point using the formula

$$\mathbf{u}_{P_s+2} = \mathbf{u}_{\text{best}}^{(g)} + \big(\mathbf{u}_{\text{best}}^{(g)} - \mathbf{u}_{Ps+1}\big).$$

3) For these two new points, compute their objective function values: $J_{i,o}$, $1 \le o \le N$ and $i = P_s+1, P_s+2$.

4) For $i = 1:2$

i) Perform **Pareto Ranking, Distance Measure and Cost Mapping** for $\{\mathbf{u}_j^{(g)}, J_{j,o}, 1 \le o \le N\}_{j=1}^{P_s+2-(i-1)}$, the enlarged population, to yields $\{\hat{J}_j\}_{j=1}^{P_s+2-(i-1)}$

ii) Find $j_{\text{worst}} = \arg \max\limits_{1 \le j \le P_s+2-(i-1)} \hat{J}_j$, and remove $\mathbf{u}_{j_{\text{worst}}}^{(g)}$ from the population

This removes the two 'worst' points, and keeps the population size to P_s.

– **End of Inner loop** Choose the P_e best solutions, $\{\mathbf{u}_{\text{best},i}^{(g)}\}_{i=1}^{P_e}$:
For $i = 1:P_e$

i) Perform **Pareto Ranking, Distance Measure and Cost Mapping** for the population $\{\mathbf{u}_j^{(g)}, J_{j,o}, 1 \le o \le N\}_{j=1}^{P_s-(i-1)}$ to yields $\{\hat{J}_j\}_{j=1}^{P_s-(i-1)}$

ii) Find $j_{\text{best}} = \arg \min\limits_{1 \le j \le P_s-(i-1)} \hat{J}_j$, set $\mathbf{u}_{\text{best},i}^{(g)} = \mathbf{u}_{j_{\text{best}}}^{(g)}$, and remove $\mathbf{u}_{j_{\text{best}}}^{(g)}$ from the population

○ **End of outer loop** This yields the solution set $\{\mathbf{u}_i^{(N_g)}\}_{i=1}^{P_s}$

III. Numerical Experiments

The performance of the proposed PRWBS algorithm was compared with the NSGA-II algorithm on several test problems. The NSGA-II is a well-known state-of-the-art multiple-objective optimisation algorithm which has been shown to produce very good results on a wide range of problems [19]. The NGSA-II algorithm used utilised real-coding, binary tournament selection, binary crossover with probability 0.9, polynomial mutation with probability $\frac{1}{n}$, and non-dominated sorting in conjunction with a crowding operator.

SCH function: The one-dimensional 'SCH' function was taken from [19], which exhibits a convex Pareto-frontier:

$$\begin{cases} J_1(u) = u^2, \\ J_2(u) = (u-2)^2. \end{cases} \tag{6}$$

The decision variable $u \in [-1,\ 1]$. The following algorithmic parameters were used for the PRWBS: $P_s = 25$, $N_B = 10$, $N_g = 100$, $P_e/P_s = 0.8$, and $P_r = 10$, which were found to produce the best results based on trial and error. The population size and the number of generations for the NSGA-II were 30 and 50, respectively, and they were also tuned using trial and error to provide the best performance.

The results obtained for this test function are illustrated in Fig. 1, which shows the resulting objective space solutions. In this figure and all the other objective-space based figures in this section, red dot markers indicate the feasible solutions generated by multiple Monte-Carlo (MC) simulations based on random sampling in the decision space which help to visually locate the Pareto-frontier, blue smaller asterisk markers indicate the candidate solutions generated by the NSGA-II, and black larger asterisk markers the candidate solutions generated by the PRWBS. It is noted from Fig. 1 that the PRWBS algorithm is capable of finding solutions across the Pareto-frontier. However, the distribution of the solutions found by the PRWBS is inferior to that of the NSGA-II, as it is less uniform across the Pareto-frontier.

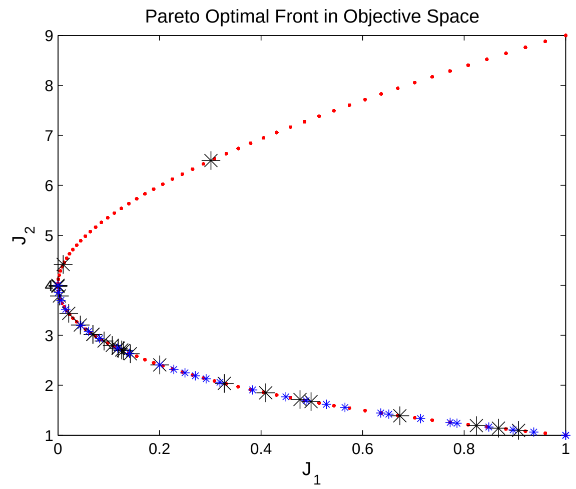

Fig. 1. Objective space performance comparison of the NSGA-II and PRWBS on the convex test problem of SCH function (6), where red dot markers indicate the feasible solutions generated by MC simulation, blue smaller asterisk markers indicate the candidate solutions generated by the NSGA-II, and black larger asterisk markers the PRWBS candidate solutions.

KUR function: The two-dimensional 'KUR' function was again taken from [19], which has a non-convex Pareto-frontier:

$$\begin{cases} J_1(\mathbf{u}) = \sum\limits_{i=1}^{n-1} -10e^{-0.2\sqrt{u_i^2+u_{i+1}^2}}, \\ J_2(\mathbf{u}) = \sum\limits_{i=1}^{n} \big(|u_i|^{0.8} + 5\sin(u_i^3)\big), \end{cases} \tag{7}$$

where $n = 2$, $\mathbf{u} = [u_1\ u_2]^{\mathrm{T}}$, $u_1 \in [-5,\ 5]$ and $u_2 \in [-5,\ 5]$. The algorithmic parameters used for the PRWBS were: $P_{\mathrm{s}} = 25$, $N_{\mathrm{B}} = 10$, $N_{\mathrm{g}} = 100$, $P_{\mathrm{e}}/P_{\mathrm{s}} = 0.8$ and $P_{\mathrm{r}} = 10$, while the population size and the number of generations for the NSGA-II were 30 and 50, respectively. These parameters were again chosen through trial and error. The results for this test function are illustrated in Figs. 2 and 3.

Again, both the NSGA-II and PRWBS algorithms focus on the the same convex region of the Pareto-frontier, as can be seen from Fig. 2. A close-up of the objective space in the region where the majority of the solutions are located is illustrated in Fig. 2 (b), which reveals that the PRWBS algorithm approaches the Pareto-frontier successfully and the solutions are distributed across a similar region as the NSGA-II candidate solutions. The distribution of the PRWBS solutions across this region, however, is less uniform in comparison with that of the NSGA-II results, as was also observed with the first test problem. The decision variable space results, plotted in Fig. 3, and all the subsequent decision variable space based figures should be interpreted as follows: the overlaid contours represent the objective functions, and blue smaller asterisk markers indicate the NSGA-II candidate solutions, while red larger asterisk markers indicate the PRWBS candidate solutions. From Fig. 3, it can be observed that the Pareto-optimal solutions lie in a very small region of the decision variable space and the PRWBS algorithm has identified a very similar region to that of the NSGA-II. However, the PRWBS solutions are slightly more spread out in the decision space in comparison with the NSGA-II solutions, indicating that there may be scopes for further improvements in the Pareto-ranking and cost adjustment process of the PRWBS.

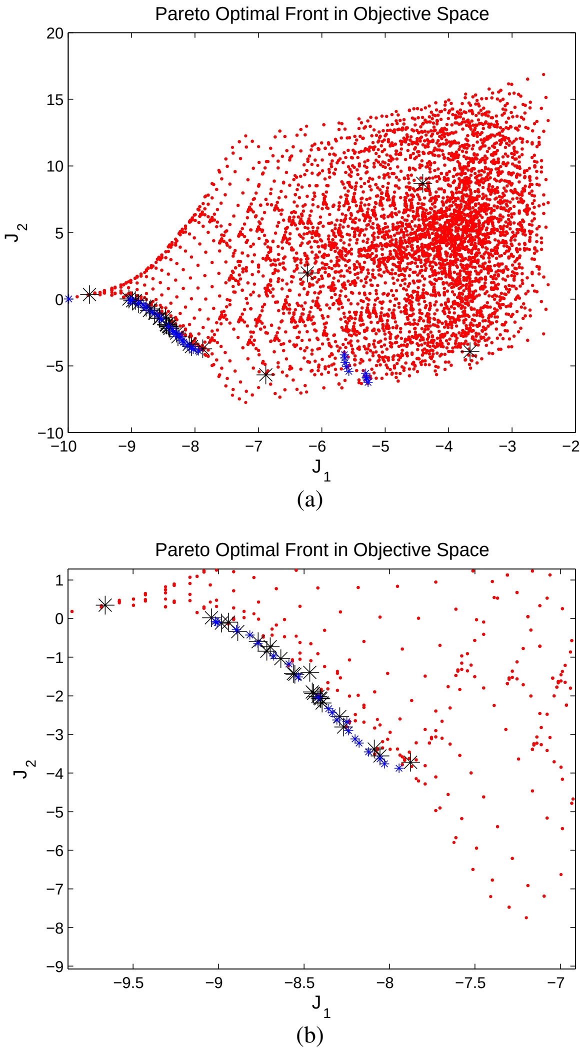

Fig. 2. Comparison of full objective space performance (a) and close-up objective space performance (b) for the NSGA-II and PRWBS on the non-convex test problem of KUR function (7), where red dot markers indicate the feasible solutions generated by MC simulation, blue smaller asterisk markers indicate the candidate solutions generated by the NSGA-II, and black larger asterisk markers the PRWBS candidate solutions.

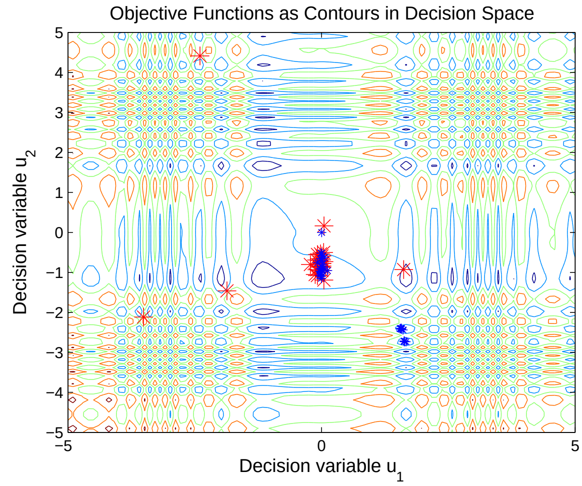

Fig. 3. Decision variable space comparison of the NSGA-II and PRWBS on the non-convex test problem of KUR function (7), where the overlaid contours represent the objective functions, blue smaller asterisk markers indicate the candidate solutions generated by the NSGA-II, and red larger asterisk markers the PRWBS candidate solutions.

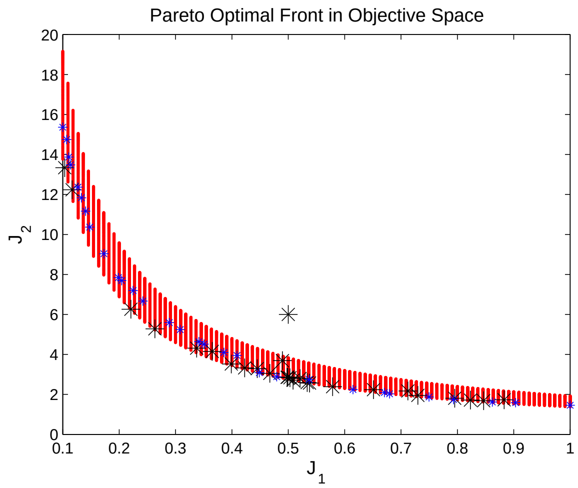

Fig. 4. Objective space performance comparison of the NSGA-II and PRWBS on the multi-modal test problem (8), where red dot markers indicate the feasible solutions generated by MC simulation, blue smaller asterisk markers indicate the candidate solutions generated by the NSGA-II, and black larger asterisk markers the PRWBS candidate solutions.

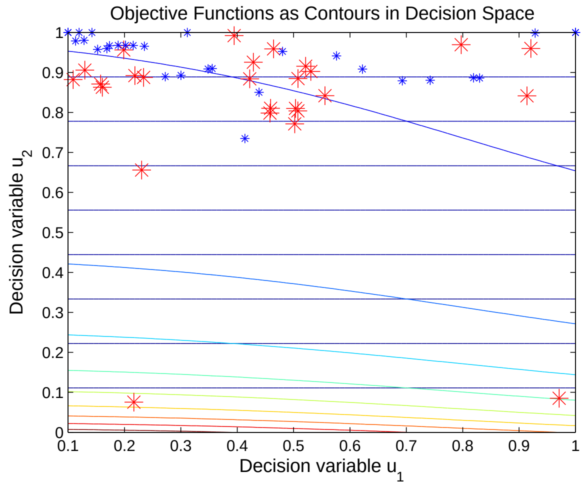

Fig. 5. Decision variable space comparison of the NSGA-II and PRWBS on the multi-modal test problem (8), where the overlaid contours represent the objective functions, blue smaller asterisk markers indicate the candidate solutions generated by the NSGA-II, and red larger asterisk markers the PRWBS candidate solutions.

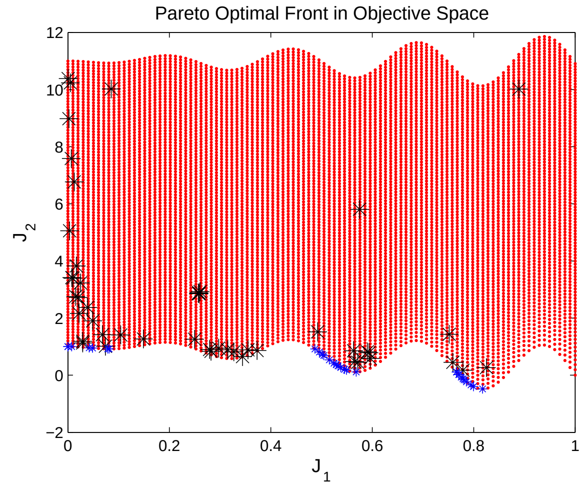

Fig. 6. Objective space performance comparison of the NSGA-II and PRWBS on the discontinuous test problem (9), where red dot markers indicate the feasible solutions generated by MC simulation, blue smaller asterisk markers indicate the candidate solutions generated by the NSGA-II, and black larger asterisk markers the PRWBS candidate solutions.

Multi-modal function: The performance of the two algorithms in the case where the Pareto-frontier is multi-modal was examined using the following two-dimensional test function adopted from [22]:

$$\begin{cases} J_1(\mathbf{u}) = u_1, \\ g(u_2) = 2.0 - e^{-\left(\frac{u_2-0.2}{0.004}\right)^2} - 0.8e^{-\left(\frac{u_2-0.6}{0.4}\right)^2}, \\ J_2(\mathbf{u}) = \frac{g(u_2)}{u_1}, \end{cases} \quad (8)$$

where $\mathbf{u} = [u_1 \; u_2]^{\mathrm{T}}$, $u_1 \in [0.1, \; 1]$ and $u_2 \in [0, \; 1]$. Again, the following algorithmic parameters were used for the PRWBS: $P_{\mathrm{s}} = 25$, $N_{\mathrm{B}} = 10$, $N_{\mathrm{g}} = 100$, $P_{\mathrm{e}}/P_{\mathrm{s}} = 0.8$ and $P_{\mathrm{r}} = 10$, while the population size and the number of generations of the NSGA-II were set to 30 and 50, respectively. The results obtained for this test function are illustrated in Figs. 4 and 5.

This optimisation problem has multiple modes, an attribute which is known to cause difficulties for many multiple-objective optimisation methods. The PRWBS algorithm demonstrates the ability to identify a range of modes and, in some regions of the Pareto frontier, outperforms the NSGA-II algorithm, as can be seen in Fig. 4. Once again, a reasonable area of the Pareto-frontier is identified by the PRWBS, but the distribution of the solutions is less uniform than that of the NSGA-II results. The relative positions of the candidate solutions in the decision variable space, as depicted in Fig. 5, are not as informative in this case, and it is difficult to infer insight into the operation of the PRWBS or NSGA-II from them. However, armed with a priori knowledge regarding the true Pareto-optimal region of the decision space, it may be possible to gain a deeper understanding, and this is an interesting area for future research.

Discontinuous function: The following two-dimensional test function is an example where the Pareto-frontier is discontinuous [22]:

$$\begin{cases} J_1(\mathbf{u}) = u_1, \\ \quad g(u_2) = 1 + 10u_2, \\ J_2(\mathbf{u}) = g(u_2)\left(1 - \left(\frac{J_1(\mathbf{u})}{g(u_2)}\right)^{\alpha} \right. \\ \qquad \left. - \frac{J_1(\mathbf{u})}{g(u_2)} \sin\left(2\pi q J_1(\mathbf{u})\right)\right), \end{cases} \quad (9)$$

where $\alpha = 2$, $q = 4$, $\mathbf{u} = [u_1 \; u_2]^{\mathrm{T}}$, $u_1 \in [0, \; 1]$ and $u_2 \in [0, \; 1]$. The algorithmic parameters, $P_{\mathrm{s}} = 50$, $N_{\mathrm{B}} = 20$, $N_{\mathrm{g}} = 100$, $P_{\mathrm{e}}/P_{\mathrm{s}} = 0.8$ and $P_{\mathrm{r}} = 10$, were found empirically for the PRWBS. The PRWBS required larger P_{s} and N_{B} for this problem, most likely due to the challenging nature of the problem. This two-objective optimisation problem has a discontinuous Pareto-frontier, an attribute which is known to challenge multiple-objective optimisation techniques. The NSGA-II used the same settings as in the previous problems.

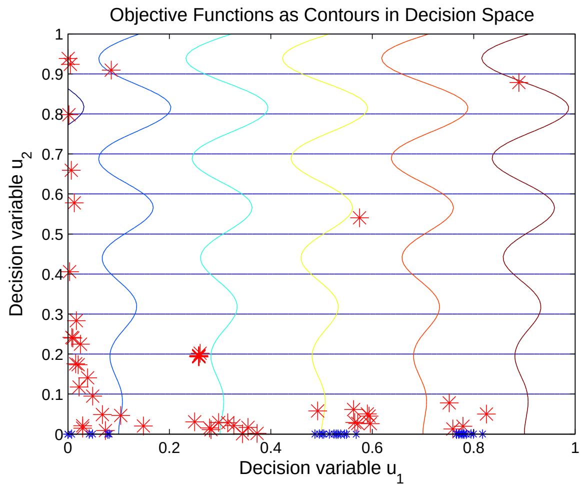

Fig. 7. Decision variable space comparison of the NSGA-II and PRWBS on the discontinuous test problem (9), where the overlaid contours represent the objective functions, blue smaller asterisk markers indicate the candidate solutions generated by the NSGA-II, and red larger asterisk markers the PRWBS candidate solutions.

The results obtained for this test function are illustrated in Figs. 6 and 7.

It is observed from Fig. 6 that the PRWBS converges towards four of the primary Pareto-optimal regions, while the NSGA-II algorithm only identifies three of the regions in this particular simulation. The performance of the NSGA-II algorithm within the three regions located by the algorithm is, however, superior to that of the PRWBS for these three regions, in terms of solution distribution. Fig. 7 also offers some intuition regarding the performance of the two algorithms. In this case, the PRWBS is observed to identify a larger area of the Pareto-frontier, in the form of four modes compared with the three modes identified by the NSGA-II, but the solutions of the PRWBS are located further from the frontier than the NSGA-II solutions.

IV. Conclusions and Future Work

A Pareto RWBS algorithm has been proposed for multiple-objective optimisation problems by providing the original single-objective RWBS algorithm with a Pareto-ranking scheme combined with a sharing process. The resulting PRWBS algorithm performs on par with the NSGA-II algorithm which is a well-known state-of-the-art multiple-objective GA, in terms of identifying the Pareto-frontier, while retaining the attractive properties of the original RWBS algorithm, namely, simplicity, ease of implementation and small number of tuning parameters.

More specifically, the PRWBS algorithm has been shown to converge reliably towards the Pareto-frontier in a range of test problems with various challenging attributes. The algorithm is observed to be capable of identifying a large area of the Pareto-frontier in each case, comparable with the NSGA-II algorithm. In particular, for the test case of discontinuous Pareto-frontier, the PRWBS provides a superior performance, in terms of locating more discontinuous regions of the Pareto-frontier. The results presented has therefore demonstrated that the PRWBS algorithm offers clear potential as a flexible, high-performance multiple-objective optimisation technique. There are scopes, however, to further improve the algorithm, both in terms of the distribution of its solutions along the Pareto-frontier, and the accuracy of the solutions in terms of their distances to the Pareto-frontier.

A future work for improving the performance of the PRWBS in this regard would be *selective combining*. The PRWBS algorithm generates a single new member by weighting all the candidates. An alternative approach would be to use a selection operator to select which members are used in a *set* of convex combinations at each stage, similar to the way a GA proceeds. This would create a number of new individuals in each generation. It is hypothesised that this approach would help to improve the algorithm's performance in terms of the accuracy to which the Pareto-frontier is located.

References

[1] S. Chen, X. X. Wang and C. J. Harris, "Experiments with repeating weighted boosting search for optimization in signal processing applications," *IEEE Trans. Systems, Man and Cybernetics, Part B*, vol. 35, no. 4, pp. 682–693, Aug. 2005.

[2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[3] K. F. Man, K. S. Tang, and S. Kwong, *Genetic Algorithms: Concepts and Design*. London: Springer-Verlag, 1998.

[4] L. Ingber, "Simulated annealing: Practice versus theory," *Mathematical and Computer Modelling*, vol. 18, no. 11, pp. 29–57, Dec. 1993.

[5] S. Chen and B. L. Luk, "Adaptive simulated annealing for optimization in signal processing applications," *Signal Processing*, vol. 79, no. 1, pp. 117–128, Nov. 1999.

[6] F. Schoen, "Stochastic techniques for global optimization: A survey of recent advances," *J. Global Optimization*, vol. 1, no. 3, pp. 207–228, 1991.

[7] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[8] S. Chen, X. X. Wang and D. J. Brown, "Orthogonal least squares regression with tunable kernels," *Electronics Letters*, vol. 41, no. 8, pp. 484–486, April 2005.

[9] S. Chen, X. X. Wang, X. Hong and C. J. Harris, "Kernel classifier construction using orthogonal forward selection and boosting with Fisher ratio class separability measure," *IEEE Trans. Neural Networks*, vol. 17, no. 6, pp. 1652–1656, Nov. 2006.

[10] X. X. Wang, S. Chen, D. Lowe and C. J. Harris, "Sparse support vector regression based on orthogonal forward selection for the generalised kernel model," *Neurocomputing*, vol. 70, nos. 1-3, pp. 462–474, Dec. 2006.

[11] S. Chen, X. X. Wang and C. J. Harris, "NARX-based nonlinear system identification using orthogonal least squares basis hunting," *IEEE Trans. Control Systems Technology*, vol. 16, no. 1, pp. 78–84, Jan. 2008.

[12] M. Abuthinien, S. Chen and L. Hanzo, "Semi-blind joint maximum likelihood channel estimation and data detection for MIMO systems," *IEEE Signal Processing Letters*, vol. 15, pp. 202–205, 2008.

[13] M. Zhang, L. Fu, G. Wang and T. He, "Improved orthogonal least-squares regression with tunable kernels using a tree structure search algorithm," *IEEE Signal Processing Letters*, vol. 15, pp. 653–656, 2008.

[14] S. A. Hoseini, R. P. Torghabeh, M. Kaveh and H. Khaloozadeh, "Designing stabilizing regulators for chaotic systems using repeated weighted boosting search method," in: *Proc. 2nd Int. Conf. Computer, Control and Communication* (Karachi), Feb. 17-18, 2009, pp. 1–6.

[15] M. Zhang, J. Zhou, L. Fu and T. He, "Hybrid wavelet model construction using orthogonal forward selection with boosting search," *Int. J. Business Intelligence and Data Mining*, vol. 3, no. 4, pp. 437–450, Jan. 2009.

[16] S. Chen, X. Hong, B. L. Luk and C. J. Harris, "Construction of tunable radial basis function networks using orthogonal forward selection," *IEEE Trans. Systems, Man, and Cybernetics, Part B*, vol. 39, no. 2, pp. 457–466, April 2009.

[17] J. Jiang and X.-P. Zhang, "A new player-enabled rapid video navigation method using temporal quantization and repeated weighted boosting search," in: *Proc. 2010 IEEE Computer Society Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)* (San Francisco, CA), June 13-18, 2010, pp. 64–71.

[18] J. Zhang, S. Chen, X. Mu and L. Hanzo, "Joint channel estimation and multi-user detection for SDMA OFDM based on dual repeated weighted boosting search," in: *Proc. ICC 2011* (Kyoto, Japan), June 5-9, 2011, 5 pages.

[19] K. Deb, A. Pratap, S. Agarwa and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NGSA-II," *IEEE Trans. Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, April 2002.

[20] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, vol. 3, no. 1, pp. 1–16, 1995.

[21] C. M. Fonseca, *Multiobjective Genetic Algorithms with Application to Control Engineering Problems*, PhD Dissertation, Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK, September 1995.

[22] K. Deb, "Multi-objective genetic algorithms – problem difficulties and construction of test problems," *Evolutionary Computation*, vol. 7, no. 3, pp. 205–230, 1999.