Pareto Repeated Weighted Boosting Search for Multiple-Objective Optimisation

Scott F. Page, Sheng Chen, Chris J. Harris
and Neil M. White

Electronics and Computer Science
Faculty of Physical and Applied Science
University of Southampton
Southampton SO17 1BJ, UK
E-mail: sqc@ecs.soton.ac.uk

11th UK Workshop on Computational Intelligence
University of Manchester, Sept. 7-9, 2011

S.F. Page is now with Waterfall Solutions Ltd, Guildford, Surrey GU2 9JX

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

Outline

1. Introduction
   - Motivations
   - Our Contributions
2. Pareto RWBS Algorithm
   - Single-Objective RWBS
   - Multiple-Objective RWBS
3. Numerical Results
   - Convex Pareto-Frontier
   - Non-convex Pareto-Frontier
   - Multi-Modal Pareto-Frontier
   - Discontinuous Pareto-frontier
4. Conclusions
   - Summary and Future Work

Outline

Global Optimisation

- Attaining **global** or near global **optimal solutions** at **affordable** computational **costs** are critical in engineering applications
- We have successful record in applications of **computational intelligence** methods, such as

  adaptive simulated annealing, genetic algorithms, ant colony / particle swarm optimisation, differential evolution algorithm
- Key metrics in assessing a method
  - **Capability**: high successful rate to attain global solutions in challenging problems
  - **Complexity**: fast convergence speed and reasonably low computational costs
  - **Simplicity**: few algorithmic parameters need tuning and easy of programming

RWBS Algorithm

- **Repeated weighted boosting search** is a **guided stochastic** search or meta-heuristic algorithm
  - Ease of implementation/programming
  - very few number of tuning parameters, and
  - capable of achieving levels of performance comparable with standard benchmark techniques, such as GA and ASA
- Successfully apply to various image and signal processing problems as well as wireless communication designs, e.g.
  - Tunable radial basis function data modelling
  - Blind joint channel estimation and data detection
  - Joint timing and channel estimation
- Original RWBS algorithm is for **single-objective** optimisation

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, 682–693, 2005.

Outline

Contributions of This Work

- Extend RWBS algorithm to **multiple-objective** optimisation
    - More specifically, arm the RWBS with a **Pareto-ranking** scheme combined with a **sharing** process
    - Similar to state-of-the-art multiple-objective GA, known as non-dominated sorting genetic algorithm (NSGA-II)
    - Resulting algorithm is therefore referred to as **Pareto** repeated weighted boosting search
- Performance of Pareto RWBS algorithm was assessed using some well-known benchmark problems
    - It offers promising level of performance in solving these multiple-objective optimisation problems,
    - while retaining the attractive properties of the original RWBS

Outline

Original RWBS

- Consider optimisation problem

$$\min_{\mathbf{u}\in\mathbb{U}^n} J(\mathbf{u})$$

$\mathbf{u} = [u_1 \ u_2 \cdots u_n]^{\mathrm{T}}$: decision variable vector, $\mathbb{U}^n$: feasible set of $\mathbf{u}$, and $J(\mathbf{u})$: cost function

- RWBS: population based guided stochastic search
  1. Stochastic search component, outer loop – "**generations**"
     - Random population initialisation with elitism
  2. Local search component, inner loop – "**weighted boosting search**"
     - Convex combination and reflection, with adaptive weighting that **boosts** weak local optimiser
- Algorithmic parameters: population size P_{s}, generations N_{g}, WBS iterations N_{B}

Algorithm

- **Outer loop**: generations for $g = 1 : N_g$
  - Random generation initialisation $\mathbf{u}_i^{(g)}$, $2 \le i \le P_s$, with elitism $\mathbf{u}_1^{(g)} = \mathbf{u}_{\text{best}}^{(g-1)}$
  - Equal initial weightings $\delta_i(0)$ and cost evaluations $J_i = J(\mathbf{u}_i^{(g)})$, $1 \le i \le P_s$
  - **Inner loop**: weighted boosting search $t = 1 : N_B$
    1. Boosting

       1) Best and worst members: $\mathbf{u}_{\text{best}}^{(g)}$ and $\mathbf{u}_{\text{worst}}^{(g)}$, according to costs $\{J_i\}$

       2) Adapt weightings $\delta_i(t)$, $1 \le i \le P_s$, according to costs $\{J_i\}$
    2. Updating

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

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

       3) Best of $\mathbf{u}_{P_s+1}, \mathbf{u}_{P_s+2}$ replaces $\mathbf{u}_{\text{worst}}^{(g)}$ in population
  - **End of Inner loop**: gth generation solution $\mathbf{u}_{\text{best}}^{(g)}$
- **End of Outer loop**: solution $\mathbf{u}_{\text{best}}^{(N_g)}$

Outline

Multiple-Objective Optimisation

- Consider optimisation problem

$$\min_{\mathbf{u}\in\mathbb{U}^n} f\big(J_1(\mathbf{u}), J_2(\mathbf{u}), \cdots, J_N(\mathbf{u})\big)$$

  $J_i(\mathbf{u})$: ith objective function, N: number of objective functions, and f: objective preference function

- True **multiple-objective** optimisation: **no objective preference** structure is available
- Set of **optimal** solutions is characterised by **Pareto-frontier**, and two key aspects of designing efficient Pareto optimisation
  1. Mechanism drives solutions toward Pareto frontier, Pareto **ranking**: promote **non-dominated** solutions
  2. Mechanism ensures distribution of solutions across Pareto frontier, **sharing**: encourage **spread** of solutions

Pareto RWBS

- **Elitism count**: to aid identifying suitable set of Pareto-optimal solutions, P_{e} population members are kept between generations
- **Pareto-ranking**: fast-non-dominated-sort procedure of (Deb *et al.*, 2002) is used to calculated Pareto-ranking

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

  where $J_{i,o} = J_o\big(\mathbf{u}_i^{(g)}\big)$, $1 \le o \le N$
- **Cost mapping**: given scaling parameter P_{r} and mean distance of $\mathbf{u}_i^{(g)}$ to other points

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

  distance and ranking **adjusted** costs

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

Algorithm

- **Outer loop**: generations for $g = 1 : N_{\text{g}}$

  – Pareto generation initialisation: $\mathbf{u}_i^{(g)} = \mathbf{u}_{\text{best},i}^{(g-1)}$, $1 \le i \le P_{\text{e}}$, and randomly generate rest of population $\mathbf{u}_i^{(g)}$, $P_{\text{e}} + 1 \le i \le P_{\text{s}}$

  – Equal initial weightings $\delta_i(0)$ and cost evaluations

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

  – **Inner loop**: weighted boosting search $t = 1 : N_{\text{B}}$

  1. Pareto Boosting
  2. Pareto Updating

  – **End of Inner loop**: choose P_{e} best solutions $\{\mathbf{u}_{\text{best},i}^{(g)}\}_{i=1}^{P_{\text{e}}}$

  For $i = 1 : P_{\text{e}}$

  i) Perform **Pareto Ranking, Distance Measure and Cost Mapping** $\{\mathbf{u}_j^{(g)}, J_{j,o}, 1 \le o \le N\}_{j=1}^{P_{\text{s}}-(i-1)} \rightarrow \{\hat{J}_j\}_{j=1}^{P_{\text{s}}-(i-1)}$

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

- **End of Outer loop**: solution set $\{\mathbf{u}_i^{(N_{\text{g}})}\}_{i=1}^{P_{\text{s}}}$

Inner Loop

1. Pareto Boosting

   1) Perform **Pareto Ranking, Distance Measure and Cost Mapping**

$$\{\mathbf{u}_i^{(g)}, J_{i,o}, 1 \le o \le N\}_{i=1}^{P_\mathrm{s}} \to \{\hat{J}_i\}_{i=1}^{P_\mathrm{s}}$$

   Find $i_\mathrm{best} = \arg \min_{1 \le i \le P_\mathrm{s}} \hat{J}_i$ and denote $\mathbf{u}_\mathrm{best}^{(g)} = \mathbf{u}_{i_\mathrm{best}}^{(g)}$

   2) Adapt weightings $\delta_i(t)$, $1 \le i \le P_\mathrm{s}$, according to costs $\{\hat{J}_i\}$

2. Pareto Updating

   1) Convex combination $\mathbf{u}_{P_\mathrm{s}+1} = \sum_{i=1}^{P_\mathrm{s}} \delta_i(t)\mathbf{u}_i^{(g)}$

   2) Reflection $\mathbf{u}_{P_\mathrm{s}+2} = \mathbf{u}_\mathrm{best}^{(g)} + (\mathbf{u}_\mathrm{best}^{(g)} - \mathbf{u}_{P_\mathrm{s}+1})$

   3) Compute $J_{i,o}(\mathbf{u}_i)$, $1 \le o \le N$ and $i = P_\mathrm{s}+1, P_\mathrm{s}+2$

   4) Removes two worst points to keep population size P_s: For $i = 1 : 2$

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

      ii) Find $j_\mathrm{worst} = \arg \max_{1 \le j \le P_\mathrm{s}+2-(i-1)} \hat{J}_j$, and remove $\mathbf{u}_{j_\mathrm{worst}}^{(g)}$

Outline

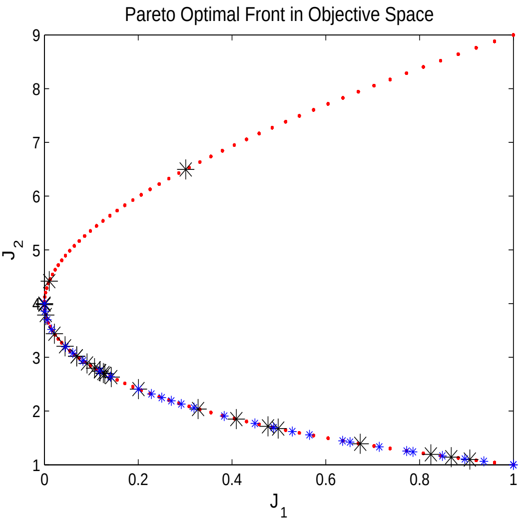
SCH Function

- One-dimensional, exhibits convex Pareto-frontier

$$J_1(u) = u^2$$
$$J_2(u) = (u-2)^2$$

$$u \in [-1,\ 1]$$

- **Red** dot: feasible solutions visualising Pareto-frontier
- **Blue** smaller asterisk: NSGA-II
- **Black** larger asterisk: Pareto-RWBS



Outline

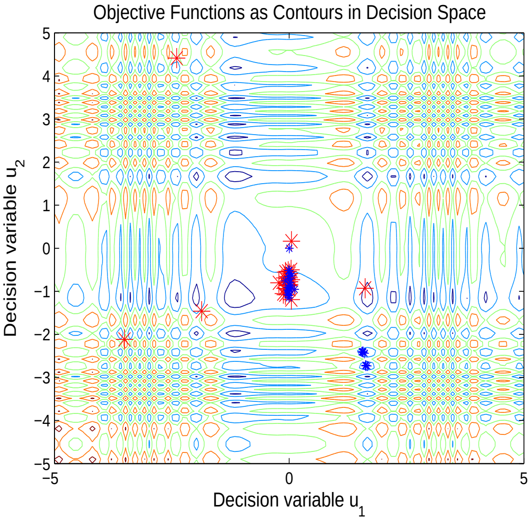
KUR Function

- Two-dimensional, exhibits non-convex Pareto-frontier

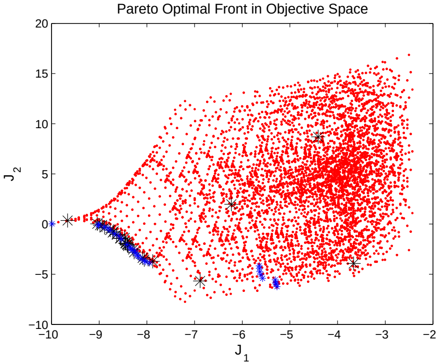
$$J_1 = -10e^{-0.2\sqrt{u_1^2+u_2^2}}$$
$$J_2 = \sum_{i=1}^{2} \left(|u_i|^{0.8} + 5\sin(u_i^3)\right)$$

$$u_i \in [-5,\ 5],\ i = 1, 2$$

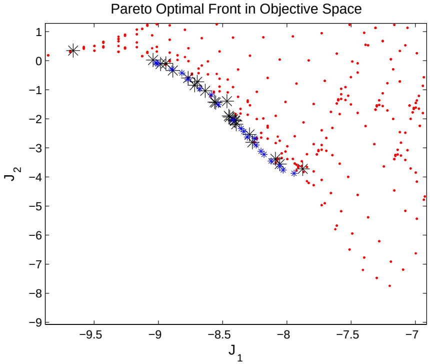
- Overlaid contours: objective functions
- **Blue** smaller asterisk: NSGA-II
- **Red** larger asterisk: Pareto-RWBS



KUR Function (continue)



(a)



(b)

(a) Full objective space, and (b) close-up objective space

- **Red** dot: feasible solutions visualising Pareto-frontier
- **Blue** smaller asterisk: NSGA-II
- **Black** larger asterisk: Pareto-RWBS

Outline

Multi-Modal Function

- Two-dimensional, difficult multi-modal Pareto-frontier

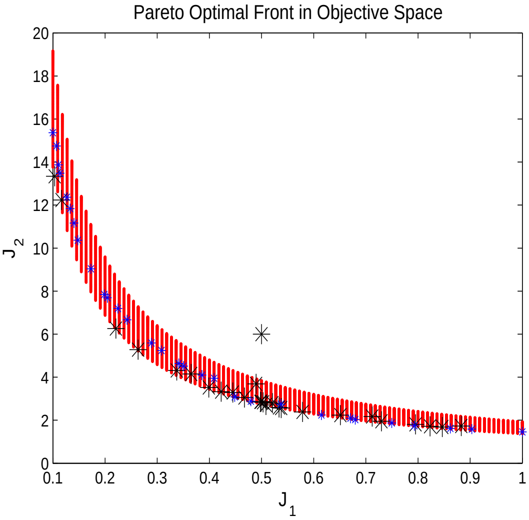
$$J_1(\mathbf{u}) = u_1$$

$$g(u_2) = 2.0 - e^{-\left(\frac{u_2 - 0.2}{0.004}\right)^2} - 0.8 e^{-\left(\frac{u_2 - 0.6}{0.4}\right)^2}$$

$$J_2(\mathbf{u}) = \frac{g(u_2)}{u_1}$$

$$u_1 \in [0.1,\ 1],\ u_2 \in [0,\ 1]$$

- **Red** dot: feasible solutions visualising Pareto-frontier
- Asterisk: **blue** smaller for NSGA-II; **black** larger for Pareto-RWBS



Outline

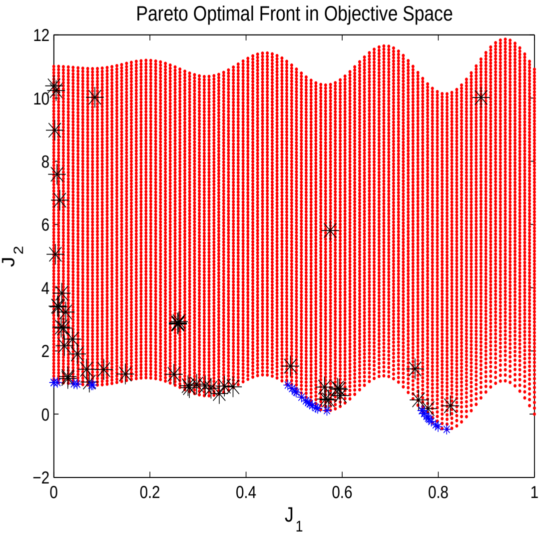
Discontinuous Function

- Two-dimensional, challenging discontinuous Pareto-frontier

$$J_1(\mathbf{u}) = u_1$$
$$g(u_2) = 1 + 10u_2$$
$$J_2(\mathbf{u}) = g(u_2)\left(1 - \left(\frac{J_1(\mathbf{u})}{g(u_2)}\right)^{\alpha} - \frac{J_1(\mathbf{u})}{g(u_2)}\sin\left(2\pi q J_1(\mathbf{u})\right)\right)$$

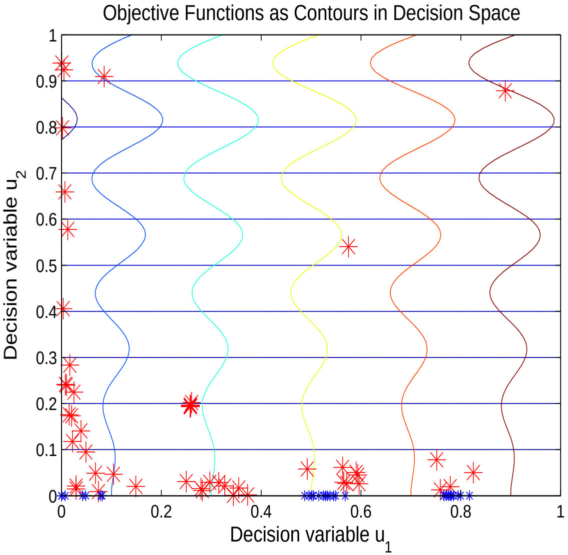
$\alpha = 2,\ q = 4,\ u_1, u_2 \in [0,\ 1]$

- **Red** dot: feasible solutions visualising Pareto-frontier
- **Blue** smaller asterisk: NSGA-II;
- **Black** larger asterisk: Pareto-RWBS



Discontinuous Function (continue)

- Overlaid contours: objective functions
- **Blue** smaller asterisk: NSGA-II
- **Red** larger asterisk: Pareto-RWBS



Outline

Summary

- Pareto RWBS algorithm for multiple-objective optimisation
  - Provide Pareto-**ranking** scheme and **sharing** process to RWBS originally for single-objective optimisation
- Pareto RWBS performs **on par** with NSGA-II algorithm
  - while retaining attractive properties: simplicity, ease of implementation and small number of tuning parameters
- Scopes to further improve Pareto RWBS:
  - improve distribution of its solutions along Pareto-frontier
  - improve accuracy of solutions in terms of their distances to Pareto-frontier

Further Work

- This Pareto RWBS generates **single** convex combination of all candidates
- Future work will investigate **selective combining**
  - develop a selection operator to select which members are used in a **set** of convex combinations
  - thus create a number of new individuals at each inner iteration
  - This is similar to the way a GA proceeds
- We hypothesise this approach will improve performance
  - in terms of solutions' distribution along Pareto-frontier and solutions' accuracy to Pareto-frontier