

PaCcET: An Objective Space Transformation to Iteratively Convexify the Pareto Front

Logan Yliniemi and Kagan Tumer

Oregon State University
Corvallis, Oregon, USA

logan.yliniemi@enr.orst.edu
kagan.tumer@oregonstate.edu

Abstract. In multi-objective problems, it is desirable to use a fast algorithm that gains coverage over large parts of the Pareto front. The simplest multi-objective method is a linear combination of objectives given to a single-objective optimizer. However, it is proven that this method cannot support solutions on the concave areas of the Pareto front: one of the points on the convex parts of the Pareto front or an extreme solution is always more desirable to an optimizer. This is a significant drawback of the linear combination.

In this work we provide the Pareto Concavity Elimination Transformation (PaCcET), a novel, iterative objective space transformation that allows a linear combination (in this transformed objective space) to find solutions on concave areas of the Pareto front (in the original objective space). The transformation ensures that an optimizer will always value a non-dominated solution over any dominated solution, and can be used by any single-objective optimizer. We demonstrate the efficacy of this method in two multi-objective benchmark problems with known concave Pareto fronts. Instead of the poor coverage created by a simple linear sum, PaCcET produces a superior spread across the Pareto front, including concave areas, similar to those discovered by more computationally-expensive multi-objective algorithms like SPEA2 and NSGA-II.

1 Introduction

Multi-objective optimization is very important in the real world [12]. Multiple competing objectives must be balanced in applications like the design of high-speed transport planes [13], the design of trusses [2], job shop scheduling [21], urban planning [1], and greywater reuse [16]. In these, the “best” solutions characterize a tradeoff between the multiple objectives. This array of solutions is known as the “Pareto optimal set”, and is a commonly sought-after solution type for a multi-objective problems [3].

Successful methods function on arbitrarily-shaped Pareto fronts, as the shape is unknown before optimization. One simple method is to use a linear combination of all objectives, which has the benefits of being easy to understand and computationally cheap, but this is unable to find the concave areas of a Pareto front [3, 4, 11, 12, 15, 17–19], because a convex part (or an extreme point) will be more desirable than the concave region [12].

The primary contribution of this work is to present the Pareto Concavity Elimination Transformation (PaCcET), a novel, optimizer-independent, iterative multi-objective transformation. It transforms the objective space so that the Pareto Front is convex, and requires only a single user-defined parameter. This allows an linear combination with unit weights (in the transformed objective space) to find concave areas of the Pareto front (in the original objective space), removing the major drawbacks of a linear combination, and allowing a simple linear combination to be used instead of more computationally expensive multi-objective evolutionary algorithms, and produce similar results.

This work is organized as follows: Section 2 provides background on multi-objective problems and multi-objective methods. Section 3 describes PaCcET. Section 4 provides theoretical guarantees for PaCcET. Sections 5 and 6 describe two test domains and show results using PaCcET. Section 7 discusses the work and concludes.

2 Background

This work draws from many distinct concepts from within multi-objective research, which we introduce in this section. We assume (without loss of generality) pure minimization of k objectives $\Lambda \in \mathbb{R}^k$ through the control of the n design variables $\Omega \in \mathbb{R}^n$.

Dominance: A solution u dominates another solution v ($u \prec v$) if it scores lower on all criteria (objectives $c \in C$): $\forall c \in C [f_c(u) < f_c(v)]$. A solution u weakly dominates another solution v ($u \preceq v$) if it scores equal on some objectives, but less on others: $\forall c \in C [f_c(u) \leq f_c(v)] \wedge \exists j \in C [f_j(u) < f_j(v)]$ [20].

Pareto optimal set: A solution which is not dominated by any other feasible solution is part of the Pareto optimal set \mathbf{P}^* . As an incomplete optimizer solves a problem, it will approximate \mathbf{P}^* with a *Pareto approximate set* P_I^* at iteration I .

Multi-objective spaces: $\Omega \in \mathbb{R}^n$ is the *design variable space* (domain). $\Lambda \in \mathbb{R}^k$ is *objective space* (range or codomain) [20]. The mapping from $\Omega \rightarrow \Lambda$ is unknown to the chosen optimizer Ξ , but is usually repeatable with some stochastic error. We also use Λ^{norm} , a normalized version of Λ , which places P_I^* elements $\in [0:1]$, and Λ^τ , the post-PaCcET analogue to Λ . Additionally, we break Λ into three sub-spaces, Λ_D , Λ_N , and Λ_B . Λ_D is the subspace in Λ that is strongly dominated by the current P_I^* . Λ_N is the subspace of Λ that is non-dominated. Λ_B forms the border between the two (Fig. 1), and includes P_I^* and all points weakly dominated by P_I^* .

Utopia and nadir vectors: Two important concepts in multi-objective problems are the utopia and nadir points. The utopia point takes on the best possible value for each objective, minus some small amount so that it is always infeasible. This point is difficult to find, requiring an optimization for each objective individually. Instead, we approximate:

$$\hat{u}^\circ(c) = \min(P_I^*(c)) - \Delta \quad (1)$$

where $\hat{u}^\circ(c)$ is the c^{th} element in the estimated utopia vector, $\min(P_I^*(c))$ is the minimum c^{th} element of any vector in P_I^* , and Δ is a small value [5]. The nadir point takes

the worst value for each objective in the Pareto optimal set, which we approximate:

$$\hat{u}^{\text{nad}}(c) = \max(P_I^*(c)) \quad (2)$$

but it is very important to note that this is a distinct concept from the worst feasible vector; it is instead the upper bound of the objective values for solutions within P_I^* [5].

2.1 Multi-Objective Methods

Many successful multi-objective algorithms have been developed. In this work we address linear combinations since they are a component of PaCcET, as well as NSGA-II and SPEA2, two successful multi-objective evolutionary algorithms.

Linear Combination: a simple metric that is sufficient, but not necessary, for finding Pareto optimal points [12]:

$$LC(w, v) = \sum_c w(c)v(c) \quad (3)$$

where $LC(w, v)$ is the linear combination evaluation or L_1 norm of vector v , $v(c)$ is the evaluation of vector v on the c^{th} objective, w is the vector of weights, and $w(c)$ is the weight for the c^{th} objective.

This method is computationally cheap when paired with typical optimizers like an evolutionary algorithm [19], but presents three primary problems. First, as the number of objectives increases, the choice of weights can become difficult. Second, this method is incapable of finding certain areas of the Pareto Front, those that are non-convex. Third, incrementing the weights evenly to converge to different parts of the Pareto front does not necessarily lead to evenly-spaced solutions along the front [4].

NSGA-II: is an evolutionary algorithm which sorts solutions into a series of successive fronts. Those solutions on the less-dominated fronts are more desirable and are kept. To break ties, a local density measure is used. Details can be found in [6].

SPEA2: is an evolutionary algorithm which assigns each vector a “strength” equal to the number of vectors in the current population it dominates. Each vector then sums the strengths of all vectors which dominate it, and this forms a raw fitness evaluation. This is altered by a local k-nearest neighbor density calculation, and the best solutions survive. Details can be found in [8].

3 Pareto Concavity Elimination Transformation (PaCcET)

Each point in the current Pareto-approximate set, P_I^* represents a tradeoff between which we are indifferent [9]. PaCcET makes each solution on A_B (including P_I^*) equally valuable to a linear combination in A^τ through a two step transformation, which first transforms from A to A^{norm} , and then transforms from A^{norm} to A^τ , where the τ superscript on any space or set denotes the transformed space or set. This means that any Pareto-approximate solution will have a linear combination evaluation of $(k - 1)$ when all weights are set to 1. All solutions in A_N^τ will have a linear combination evaluation $< (k - 1)$, and all solutions in A_D^τ will have a linear combination evaluation $> (k - 1)$.

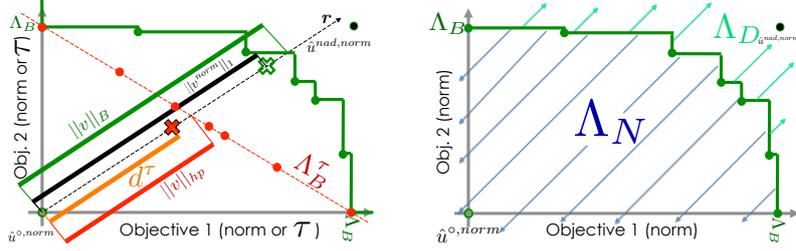


Fig. 1. Visualization of quantities used in transformation (left) and partitions in the multi-objective space (right). The vector v^{norm} is represented by the hollow green X mark, and v^τ by the solid red X mark. v^{norm} lies outside of the dominated hypervolume, so is a desirable point to discover. Green dots correspond to vectors in P_I^* (which form the border, Λ_B , between the non-dominated hyperspace Λ_N and the dominated hyperspace, Λ_D). Red correspond to their transformations in P_I^* . All measurements are Manhattan Distance (L_1 norm) along r .

Algorithm: To determine the transformed evaluation for a given solution vector v , we require the current Pareto approximate set P_I^* , from which we can calculate the approximate utopia point \hat{u}° based on P_I^* (Eq. 1), and the matching nadir approximation \hat{u}^{nad} (Eq. 2).

The first step is to normalize the target vector v such that each objective takes on a value not less than 0, transforming Λ to Λ^{norm} [5, 12]:

$$v^{\text{norm}}(c) = \frac{v(c) - \hat{u}^\circ(c)}{\hat{u}^{\text{nad}}(c) - \hat{u}^\circ(c)} \quad (4)$$

By definition $\hat{u}^{\text{nad, norm}} \equiv \mathbf{1}$ and $\hat{u}^{\circ, \text{norm}} \equiv \mathbf{0}$, and each element of a member of P_I^* will be in the range [0:1].

The second step is to perform the transformation from Λ^{norm} to Λ^τ . Within this process, we use the unit vector r that points from $\hat{u}^{\circ, \text{norm}}$ toward v^{norm} :

$$r = \frac{v^{\text{norm}}}{|v^{\text{norm}}|} \quad (5)$$

All distance measurements in the transformation process are taken along the direction of r . We measure three distances for use in PaCcET:

Algorithm 1 PaCcET for iteration I

Require: Set of solutions \mathbf{V}	6: Find $\ v^{\text{norm}}\ _1$ (Eq. 6)
Require: Pareto Approximate Set P_I^*	7: Find $\ v\ _B$ (Eq. 7)
1: Find $\hat{u}^\circ, \hat{u}^{\text{nad}}$ (Eq. 1–2)	8: Find $\ v\ _{\text{hp}}$ (Eq. 8)
2: $\forall c w_c = 1$	9: Find d^τ (Eq. 9)
3: for all Solutions $i \in \mathbf{V}$ do	10: Find v_i^τ (Eq. 10)
4: Find v_i^{norm} (Eq. 4)	11: $Fit_{\text{PaCcET}}(v_i) = LC(v_i^\tau)$ (Eq. 3)
5: Find r (Eq. 5)	12: end for

- L_1 distance (linear combination or Manhattan Distance) from $\hat{u}^{\circ, norm}$ to v^{norm} :

$$\|v^{norm}\|_1 = \sum_i v_i^{norm} \quad (6)$$

- L_1 distance from $\hat{u}^{\circ, norm}$ to the normalized dominated border Λ_B^{norm} along \mathbf{r} :

$$\|v\|_B = \min(\gamma) \ni \gamma \mathbf{r} \succeq P_I^* \quad (7)$$

- L_1 distance from $\hat{u}^{\circ, norm}$ to the normalized utopia hyperplane Λ_B^r [14] along \mathbf{r} :

$$\|v\|_{hp} = \beta \ni \sum_i \beta \mathbf{r}_i = (k - 1) \quad (8)$$

We then calculate d^τ , which determines where v^τ is located:

$$d^\tau = \|v\|_{hp} \frac{\|v^{norm}\|_1}{\|v\|_B} \quad (9)$$

And finally we determine the location of v^τ , enclosing the whole process:

$$v^\tau = d^\tau \mathbf{r} = PaCcET(v) \quad (10)$$

Choosing the Maximum Size of P_I^ , the Pareto approximate set:* P_I^* is maintained in the same way as any Pareto optimality calculation. However, for computation and memory concerns, its size must be limited. The size of P_I^* is the only user-defined parameter in PaCcET, and corresponds directly to the granularity of the Pareto front estimation. In our experiments we use 250 as the size. We ran tests with a size as small as 50, in which the algorithm still functions, but provides a very coarse approximation of the true Pareto front. Once over the chosen size, we used random elimination of non-extreme elements. We also tested with nearest-neighbor elimination and k -nearest neighbor elimination, the performance of PaCcET was not sensitive.

4 Theoretical Properties of PaCcET

In this section we provide two theorems which together prove that PaCcET finds Pareto optimal solutions, even in concave areas of the Pareto front. We begin by assuming:

Assumption A1 *The system designer specifies k points that are incomparable to the Pareto front, which describe a hyper-prism that completely bounds the Pareto Front.*

Assumption A2 *Optimizer Ξ solves the PaCcET problem exactly in a single iteration.*

Assumption A3 *The feasible region has no solutions that are weakly dominated by the Pareto Front.*

Assumption A4 *The Pareto Front is continuous.*

A1 provides us vectors with which we seed P_I^* , and assures PaCcET is calculable in the whole feasible objective space. A2 allows us to use the exact solution to the PaCcET minimization problem to determine how P_I^* changes over iterations. A3 and A4 allow us to draw conclusions in k -objective space without any other restrictions on the shape of the Pareto Front.

Theorem 1 *The solution to the PaCcET optimization problem will be Pareto Optimal.*

Proof. There exists an infinite number of possible rays $\mathbf{r} \in \mathbf{R}$ (where \mathbf{R} is the set of all rays originating from $\mathbf{0}$) on which the true solution may exist. This solution exists only along one of those rays, which must pass through the feasible space. We do not seek to determine which \mathbf{r} it lies on. For any individual \mathbf{r} , the PaCcET optimization problem takes the form (Eq. 9, reorganized):

$$\min(d^\tau) = \min \left(\|v\|_{\text{hp}} \frac{\|v^{\text{norm}}\|_1}{\|v\|_{\text{B}}} \right) \quad (11)$$

And for a constant \mathbf{r} , $\|v\|_{\text{B}}$ and $\|v\|_{\text{hp}}$ are constant on a given iteration:

$$\min(d^\tau) = \min(\alpha \|v^{\text{norm}}\|_1) \quad (12)$$

where α is some positive constant. $\|v^{\text{norm}}\|_1$ increases monotonically as distance from the origin increases, therefore d^τ does as well. The minimum of d^τ , then, will be on the border of the feasible space, a Pareto Optimal Solution or a weakly dominated solution [12]. By A3 and A4, this is a Pareto optimal solution. This can also be assured by the same logic as [4], since it is equivalent to a scaled linear combination. \square

Theorem 2 *PaCcET finds solutions in concave areas of the Pareto front.*

Proof. We prove this by contradiction. Assume a globally concave search space. By theorem 1, in the worst case, the solution to the PaCcET optimization problem will lead to the k anchor points (single objective extremes) in the first k iterations. By A4, we know additional Pareto optimal points exist. We show that the d^τ calculations for those points in the current P_I^* is greater than those in A_N (super/sub-scripts denoting the calculation for a member of the set named in the super/sub-script):

$$d_{P_I^*}^\tau > d_{A_N}^\tau \quad (13)$$

$$\|v\|_{\text{hp}}^{P_I^*} \frac{\|v_{P_I^*}^{\text{norm}}\|_1}{\|v\|_{\text{B}}^{P_I^*}} > \|v\|_{\text{hp}}^{A_N} \frac{\|v_{A_N}^{\text{norm}}\|_1}{\|v\|_{\text{B}}^{A_N}} \quad (14)$$

$$(k-1) \frac{\|v_{P_I^*}^{\text{norm}}\|_1}{\|v\|_{\text{B}}^{P_I^*}} > (k-1) \frac{\|v_{A_N}^{\text{norm}}\|_1}{\|v\|_{\text{B}}^{A_N}} \quad (15)$$

By definition $\|v\|_{\text{hp}}^{P_I^*} = (k-1)$. Also, $\frac{\|v_{P_I^*}^{\text{norm}}\|_1}{\|v\|_{\text{B}}^{P_I^*}} = 1$ because $\|v_{P_I^*}^{\text{norm}}\|_1 = \|v\|_{\text{B}}^{P_I^*}$, and the quantity $\frac{\|v_{A_N}^{\text{norm}}\|_1}{\|v\|_{\text{B}}^{A_N}} \in [0 : 1)$, because it is in the non-dominated subspace (so $\|v_{A_N}^{\text{norm}}\|_1 < \|v\|_{\text{B}}^{A_N}$), and the inequality in Eq. 15 holds. Because of Theorem 1, we know that the solution will be Pareto Optimal, and because of the globally concave assumption, we know this point is on a concave region of the Pareto front. \square

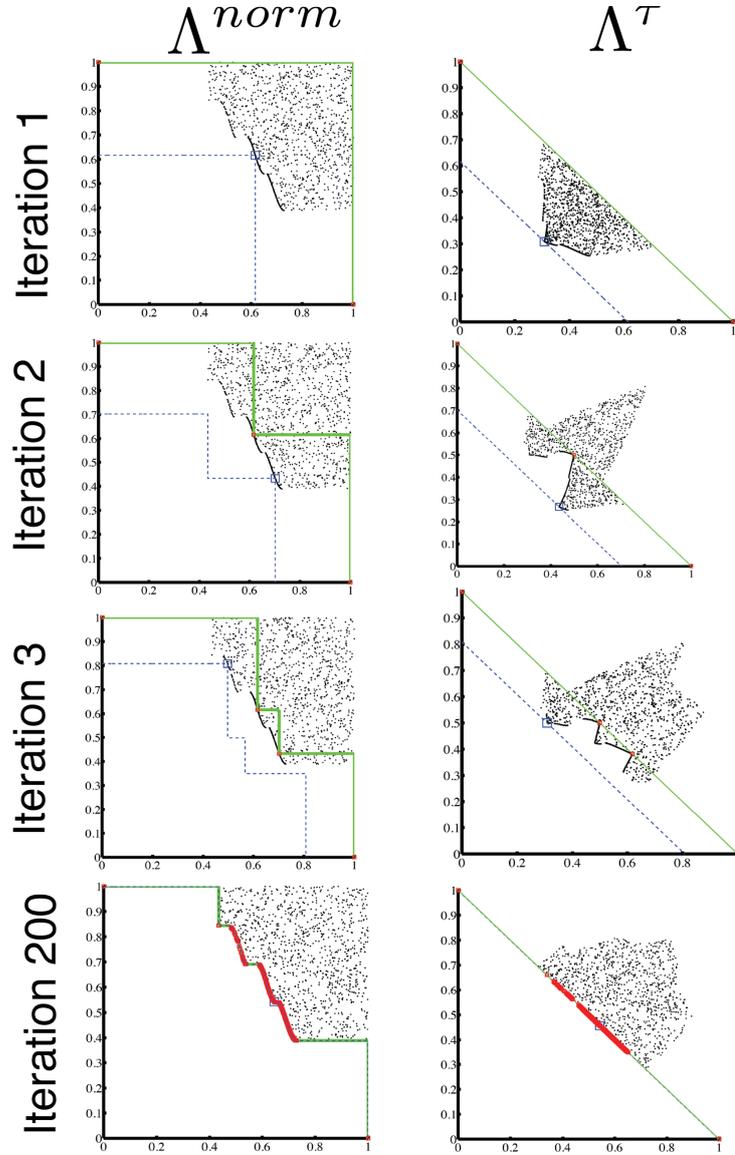


Fig. 2. Visualization of PaCcET procedure over iterations. The left column is the normalized objective space Λ^{norm} . The right column is the transformed objective space, Λ^τ . The rows show, in turn, the optimizer working at the 1st, 2nd, 3rd, and 200th iteration. In the left column, Black points are candidate solutions. Red points are solutions in P_I^* , the Pareto approximate set. The green solid line denotes Λ_B^{norm} . The blue square denotes the true solution to the PaCcET optimization problem at that iteration. The blue dashed line is the level curve of the PaCcET evaluation on which all solutions are as valuable as the discovered solution. In the right column, the colors and symbols map to the transformed versions of the same points as described previously, in Λ^τ .

Implications: The significance of these two theorems is as follows: the true solution to the PaCcET problem will always be a Pareto optimal solution, and PaCcET will be able to find concave areas of the Pareto front. Because the assumptions used to generate these conclusions are restrictive, in the following empirical results sections, we take steps to violate each of the assumptions categorically, and PaCcET is still able to find good coverage over concave Pareto fronts.

5 Experiment: KUR

As a first experimental domain, we use a test problem (KUR) from multi-objective optimization with a discontinuous and locally concave Pareto front (which breaks A4) [5]:

$$f_1(\mathbf{x}) = \sum_{i=1}^2 \left[-10 \exp \left((-0.2) \sqrt{x_i^2 + x_{i+1}^2} \right) \right] \quad (16)$$

$$f_2(\mathbf{x}) = \sum_{i=1}^3 [|x_i|^{0.8} + 5 \sin(x_i)^3] \quad (17)$$

Where f_1 and f_2 are to be minimized by controlling the decision variables:

$$x_i \in [-5, 5] \quad ; \quad i \in \{1, 2, 3\} \quad (18)$$

A vector \mathbf{x} is evaluated:

$$Fit_{LC}(\mathbf{x}) = w_1 f_1 + w_2 f_2 \quad (19)$$

where in this experiment, $w_1 = w_2 = 1$ (other values lead to different portions of the Pareto front being better covered, but similar overall performance). For PaCcET:

$$Fit_{PaCcET}(\mathbf{x}) = f_1^\tau + f_2^\tau \quad (20)$$

where f_1^τ and f_2^τ represent the transformed objectives, within Λ^τ , calculated as:

$$\{f_1^\tau, f_2^\tau\} = PaCcET(\{f_1, f_2\}) \quad (21)$$

As the optimizer Ξ , we use an evolutionary algorithm (which breaks A2), in which the population members are vectors of length 3 that meet the criteria set forth in Eq. 18. We maintain a population of 100 solutions, with the 50 worst-performing solutions removed after each generation, replaced by copies of the winner of 50 binary tournaments, with each element of the vector changed by a random number chosen by a normal distribution centered around 0 with standard deviation 0.25. We do not seed P_I^* (which breaks A1).

Figure 3 shows the Empirical Attainment Function (EAF) [10] for each method, respectively. It shows PaCcET's worst performance exceeds that of the linear combination's median performance, and PaCcET's worst performance exceeds NSGA-II's worst performance. SPEA2 and PaCcET perform comparably after 5000 generations.

Figure 4 shows the percent of dominated hypervolume by PaCcET and two successful multi-objective methods, SPEA2 and NSGA-II, as a function of number of individual fitness evaluations. PaCcET proceeds faster than the other two methods toward the Pareto front. All methods shown eventually converge to a good approximation of the Pareto front, and dominate a similar amount of hypervolume.

6 Experiment: DTLZ2

As a second experimental domain, we use one of the test problems out of the battery developed by Deb, Thiele, Laumanns and Zitzler, DTLZ2 [7]. A solution is described by a vector $(\mathbf{x} = \{x_1, x_2, \mathbf{x}_M\})$ of length 12, where 2 elements (x_1, x_2) determine at what angles in the 3 dimensional objective space evaluation v will lie and the remaining 10 elements (\mathbf{x}_M) determine the distance from the origin at which v will lie. The three functions to be minimized are:

$$f_1(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos\left(x_1 \frac{\pi}{2}\right) \cos\left(x_2 \frac{\pi}{2}\right) \quad (22)$$

$$f_2(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos\left(x_1 \frac{\pi}{2}\right) \sin\left(x_2 \frac{\pi}{2}\right) \quad (23)$$

$$f_3(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \sin\left(x_1 \frac{\pi}{2}\right) \quad (24)$$

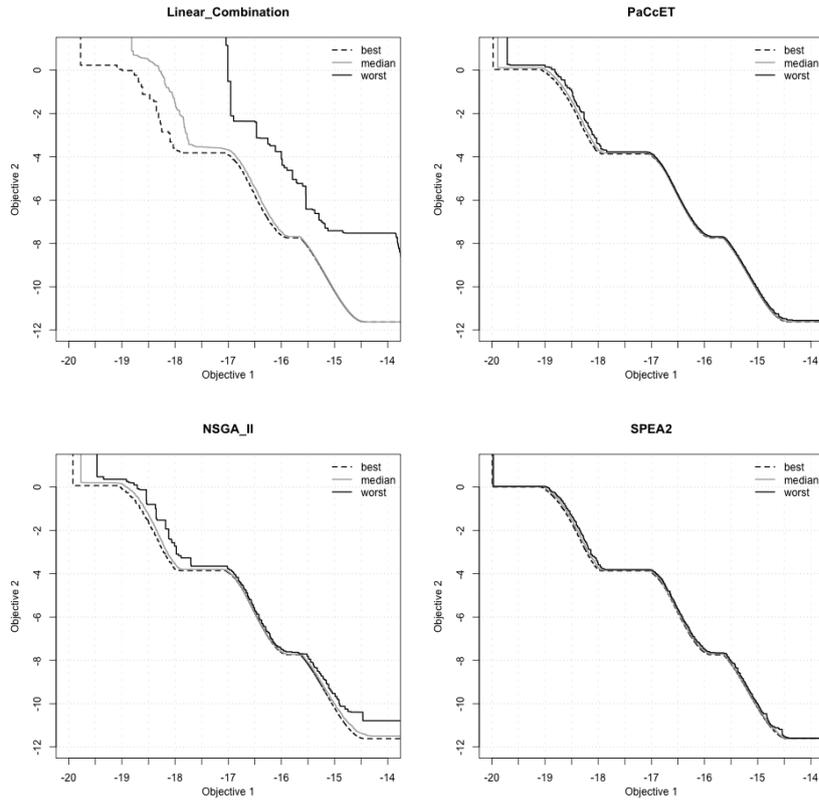


Fig. 3. KUR Empirical Attainment Functions, shown in \mathcal{A} .

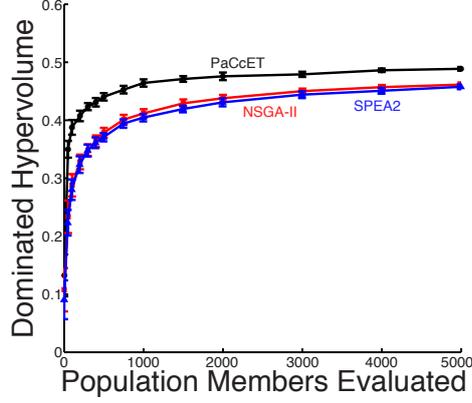


Fig. 4. Percent of hypervolume dominated in KUR, calculated using the limits of Fig. 3 over 50 statistical runs. Error in the mean (σ/\sqrt{N} where $N = 50$), is smaller than the plotted symbols.

subject to each element of \mathbf{x} remaining in the range $[0:1]$, and the evaluation $g(\mathbf{x}_M)$ calculated as:

$$g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2 \quad (25)$$

This results in a known Pareto front that can be described by the octant of a sphere of radius 1 for which f_1, f_2 , and f_3 are all positive. The feasible space has a large area that is weakly dominated by the Pareto front (which breaks A3).

The fitness of a vector \mathbf{x} is calculated as:

$$Fit_{LC}(\mathbf{x}) = w_1 f_1^{\text{norm}} + w_2 f_2^{\text{norm}} + w_3 f_3^{\text{norm}} \quad (26)$$

and for PaCcET,

$$Fit_{PaCcET}(\mathbf{x}) = f_1^\tau + f_2^\tau + f_3^\tau \quad (27)$$

where f_1^τ, f_2^τ , and f_3^τ represent the transformed objectives, within Λ^τ , calculated as:

$$\{f_1^\tau, f_2^\tau, f_3^\tau\} = PaCcET(\{f_1, f_2, f_3\}) \quad (28)$$

We use the same optimizer Ξ for DTLZ2 as for KUR (which breaks A2), except members are vectors of length 12 with each element in the range $[0:1]$, and the mutation

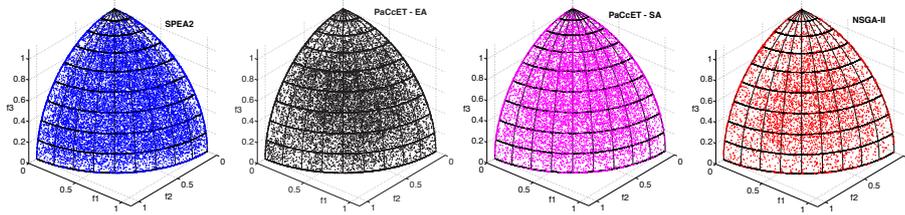


Fig. 5. All Pareto optimal points discovered in one statistical run of DTLZ2 in A .

operator alters each element by a random number drawn from a normal distribution centered around 0 with standard deviation 0.05. We do not seed P_I^* (which breaks A1).

Figure 5 shows the results on DTLZ2 for a typical experimental run of 5000 generations for each method (simulated annealing allowed the same number of global function calls as the EAs), reporting the non-dominated points found through the entire experimental run (Note that this is distinct from P_I^* , which was kept at a size of 250). SPEA2 and PaCcET using an evolutionary algorithm (PaCcET – EA) both find a similar number of solutions spread all across the Pareto front. PaCcET using simulated annealing (PaCcET – SA) is slightly less successful but still generates good coverage, even though it is not using a population-based optimizer. NSGA-II produces fewer Pareto optimal points, but still maintains coverage. The linear combination (not shown) converges to one of the extremes very quickly, producing very poor coverage, regardless of the choice of weights.

7 Discussion and Conclusion

In this work we have presented a low computational cost way to improve the performance of a linear combination in multi-objective problems. PaCcET convexities concave regions of the Pareto front for the sake of training, and allows for solutions in these areas to be found by an optimizer using a linear combination of transformed objectives.

The primary benefits of PaCcET displayed in this work are:

1. It allows a linear combination of transformed objectives to find concave areas of the Pareto front in the original objective space.
2. It acts independently of the chosen optimizer.
3. It creates a wide spread of solutions along the Pareto front on concave or discontinuous fronts.
4. It removes the need for the system designer to choose weights.
5. It functions in higher-than-two objective problems.

The first benefit (1) allows a simple linear combination to be applied to a much broader class of multi-objective problems than it could be otherwise. Benefit (2) means that optimizers like evolutionary algorithms, A* search, simulated annealing, or particle swarm optimization can be applied to multi-objective problems through PaCcET with little alteration; it also means that future developments in single-objective optimizers are immediately useful to a large class of multi-objective problem, but comes at the cost that PaCcET is limited by the quality of the optimizer. Benefit (3) reinforces (1): Even on challenging Pareto fronts, PaCcET develops a desirable array of solutions to choose between. Benefits (4,5) remove one of the primary challenges in using a linear combination on more-than-two objective problems.

PaCcET offers a fundamentally different possible avenue for multi-objective research: the elimination of concavity as opposed to the development of methods that deal well with concave Pareto fronts. Future work in this area includes testing the PaCcET on a large testbed of multi-objective problems including many-objective problems, examining how reference points can be used in the τ -objective space to steer the search or otherwise altering the method so that it is steerable, and developing guarantees for complete Pareto front coverage.

8 Acknowledgements

This work was partially supported by the National Energy Technology Laboratory under grants DE-FE0011403 and DE-FE0012302.

References

1. R. Balling, J. Taber, M. Brown, and K. Day. Multiobjective urban planning using genetic algorithm. *Journal of Urban Planning and Development*, 125(2):86–99, 1999.
2. C. A. Coello and Alan D. Christiansen. Multiobjective optimization of trusses using genetic algorithms. *Computers and Structures*, 75(6):647–660, 2000.
3. C. A. C. Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1(3):269–308, 1999.
4. I. Das and J. E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural Optimization*, pages 63–69, 1997.
5. K. Deb. *Search Methodologies*, chapter 10, pages 273–316. Springer, 2005.
6. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast elitist multi-objective genetic algorithm: NSGA-II. *Evolutionary Computation*, 6:182–197, 2002.
7. K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. Technical report, ETH Zurich, 2001.
8. M. Laumanns E. Zitzler and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. *Computer Engineering*, 3242(103), 2001.
9. Francis Ysidro Edgeworth. *Mathematical Psychics: An essay on the application of mathematics to moral sciences*. C. Kegan Paul and Company, 1881.
10. C. M. Fonseca, A. P. Guerreiro, M. Lopez-Ibanez, and L. Paquete. On the computation of the empirical attainment function. *Lecture Notes in Computer Science*, 6576:121–135, 2011.
11. R. T. Marler and J. S. Arora. The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization*, 2009.
12. R.T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26:369–395, 2004.
13. A. Messac and P. D. Hattis. Physical programming design optimization for high speed civil transport (hsct). *Journal of Aircraft*, 33(2):446–44, March 1996.
14. A. Messac, A. Ismail-Yahaya, and C. A. Mattson. The normalized normal constraint method for generating the pareto frontier. *Struct. and Multidisc. Optimization*, 25:86–98, 2003.
15. K.E. Parsopoulos and M. N. Vrahatis. Particle swarm optimization method in multiobjective problems. *ACM Symposium on Applied Computing*, 2002.
16. R. Penn, E. Friedler, and A. Ostfeld. Multi-objective evolutionary optimization for greywater reuse in municipal sewer systems. *Water research*, 47(15):5911–592, 2013.
17. W. Rosehart, C. A. Cañizares, and V. H. Quintana. Multi-objective optimal power flows to evaluate voltage security costs in power networks. *IEEE Tr. on Power Systems*, 2001.
18. P. Vamplew, R. Dazeley, A. Berry, R. Issabekov, and E. Dekker. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine Learning*, 2010.
19. David A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications Analyses and New Innovations*. PhD thesis, Air Force Institute of Technology, 1999.
20. David A. Van Veldhuizen and Gary B. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation*, 8(2):125–147, 2000.
21. G. Zhang, X. Shao, P. Li, and L. Gao. An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. *Computers and Industrial Engineering*, 56:1309–1318, 2009.