The Combative Accretion Model – Multiobjective Optimisation Without Explicit Pareto Ranking

Adam Berry and Peter Vamplew

School of Computing, University of Tasmania, Private Bag 100,
Hobart, Tasmania, Australia
E-mail: {Adam.Berry, Peter.Vamplew}@utas.edu.au

Abstract. Contemporary evolutionary multiobjective optimisation techniques are becoming increasingly focussed on the notions of archiving, explicit diversity maintenance and population-based Pareto ranking to achieve good approximations of the Pareto front. While it is certainly true that these techniques have been effective, they come at a significant complexity cost that ultimately limits their application to complex problems. This paper proposes a new model that moves away from explicit population-wide Pareto ranking, abandons both complex archiving and diversity measures and incorporates a continuous accretion-based approach that is divergent from the discretely generational nature of traditional evolutionary algorithms. Results indicate that the new approach, the Combative Accretion Model (CAM), achieves markedly better approximations than NSGA across a range of well-recognised test functions. Moreover, CAM is more efficient than NSGAII with respect to the number of comparisons (by an order of magnitude), while achieving comparable, and generally preferable, fronts.

1 Introduction

As the artificial intelligence community realises the importance of multiobjective optimisation in real-world problem domains, research attention continues to grow, with a majority of the effort being focussed on the development and investigation of Multi-Objective Evolutionary Algorithms (MOEA) [1]. At the core of much of this research rests Pareto-ranking – a concept that has been prevalent since Goldberg's early work [2] and features in a host of techniques (such as NSGA [3], MOGA [4], NSGAII [5], SPEA [6] and SPEAII [7]). Such popularity is grounded on the assumption that "Pareto ranking is the most appropriate way to generate an entire Pareto front" [1], and results investigating its use certainly support such a theory. However, despite garnering both popularity and legitimately impressive results, Pareto-ranking is not without considerable limitations. The approach carries a significant complexity cost due to its reliance on population-wide comparisons and is typically accompanied by a diversity controlling parameter that is both difficult to tune and generally expensive to use. Such is the level of computational burden that populations are fundamentally limited in size and the potential for MOEA use in high-dimensional or difficult real-world problem domains is restricted.

While a minority of contemporary algorithms endeavour to address the complexities introduced by the Pareto-ranking approach (see section 2.1), most second generation MOEA techniques extend the procedure through the inclusion of archiving, elitism and minor variations in the selection procedure (such as SPEA, SPEAII, PAES [8] and, to a lesser extent, PESA [9] and PESAII [10]). The inference that can be drawn from such a trend is that archiving and elitism are beneficial inclusions for general MOEA design, though results supporting such a claim are limited and lacking theoretical rigour. Moreover, given that the inclusion of an active secondary population generally incurs increased complexity, it is worth considering that such archiving need not be a pre-requisite for contemporary MOEA systems at all.

Consequently, this paper presents a model that moves away from active-archiving, while adopting an adaptable, inexpensive and implicit Pareto-ranking scheme that is grounded in pair-wise comparisons and simple diversity control. Furthermore, the population life-cycle is continuous rather than discrete (akin to Artificial Life systems) and agent generation is largely accretion – based on a consolidation of genes from an adaptive gene pool. Thus, the Combative Accretion Model (CAM) proposed herein represents a particularly novel approach to MOEA design that focuses on reducing complexity whilst maintaining high levels of performance.

2 Background

2.1 Pareto Ranking

Since the aim of all multiobjective optimisers is to develop an approximation of the Pareto optimal front, it is not particularly surprising that both contemporary and traditional efforts largely favour population-based Pareto dominance as a measure of fitness. By promoting those solutions that are non-dominated with respect to the current population, selection pressure favours exploration of potentially promising areas of the search space and focuses investigation on the current non-dominated front.

While it is apparent that measuring Pareto dominance is valuable in determining the direction of search, it is also significantly expensive. Even in the simplest case, where the population is divided into just two classes, the complexity[1] is $O(n^2)$ and infers a limiting bound on feasible population sizes. Such expense is only exacerbated as ranking becomes more fine-grained and continuous subdivision of dominated fronts is required (as in NSGA and NSGAII).

The complexities inherent in population-wide ranking have led to a number of algorithmic alternatives in the literature. Perhaps the most obvious approach, and the one adopted by Horn and Nafpliotis [11], is to reduce the percentage of the population under consideration when assessing dominance. In the Niched-Pareto Genetic Algorithm (NPGA), a tournament selection procedure is used, where the victor is determined by a single layer ranking process based on only ten percent of the total population (with ties broken through diversity estimation). By limiting the size of the

[1] This paper measures complexity in terms of the number of solution comparisons per evaluation (as per [1]) – objective comparisons are an equally valid measure and can be obtained by increasing comparison complexity by a factor of k, where k is the number of objectives.

population used, NPGA can gain increases in efficiency of up to an order of magnitude, while still capitalising on the general features of simple ranking. However, since the ratio of the selected population is statically defined, the process is inflexible – unable to adapt when more fine-grained analysis is required and less likely to encourage the exploration of poorly populated, but highly beneficial fronts. Moreover, results indicate that the overall quality of NPGA produced fronts is considerably worse than those utilising complete population sets [12].

A differing, though similarly motivated, approach is offered in the Pareto Archived Evolution Strategy (PAES), which considers only pair-wise dominance between parent and offspring until incomparability forces single-layered ranking against an archived set. While certainly promising, PAES suffers from its hill climbing characteristics – with the potential for significant performance degradation in problems featuring large local optima and disconnected fronts.

More recently, work has commenced on improving the efficiency of Pareto ranking by analysing the naïve list-based storage and linear search methods used in conventional MOEA. By incorporating modified versions of pre-existing efficient data structures and search algorithms, Jensen [13] outlines improvements for a host of MOEA and focuses particularly on the popular NSGAII. Although promising, and certainly worthy of continued research focus, results are minimal and suggest that tangible improvements are most noticeable with a reasonably small number of objectives. Moreover, irrespective of results, the development of more efficient structural representations and sorting methodologies should not preclude the refinement or extension of MOEA algorithms – efficiency gains in either area are likely to be of a complimentary nature and can only benefit the applicability of multiobjective optimisers in real world problem domains.

Given that approaches which endeavour to reduce the impact of ranking have only met with limited success, it is surprising that more MOEA research has not endeavoured to abandon its use altogether. The Artificial Life community has placed some focus on this concept, limiting comparisons in predator-prey systems to strictly pair-wise procedures [14, 15] and abandoning the use of dominance entirely in plant-based algorithms [16]. While such approaches typically induce significant reductions in complexity, results are of a strictly preliminary nature and require further investigation before gaining widespread acceptance.

Thus, Pareto ranking simultaneously represents the impetus for both performance efficacy and for efficiency degradation – it is the double-edged sword of multiobjective optimisation. Until the corresponding performance issues are addressed and effectively dealt with – be it through structural representation, improved search techniques or algorithmic refinement – the cost of high complexity will inevitably loom large over multiobjective optimisers in practical domains. It is not enough for researchers to focus simply on end results any longer – the utility of Pareto ranking has long since been known – the key now is to achieve those end results *efficiently*.

2.2 Complex Diversity Preservation

Where Pareto ranking explicitly guides the population towards the Pareto front, diversity preservation techniques are charged with ensuring that solutions remain well distributed along that front. By capitalising on techniques such as fitness sharing, diversity preservation reduces the likelihood of genetic drift and aids in developing a better picture of the true shape of the Pareto optimal region.

Although diversity preservation is generally a secondary operation, the complexity costs incurred through its inclusion can be as high as fitness assignment [13] and must therefore be considered prohibitively expensive. Such efficiency degradation is particularly evident in those algorithms that are reliant on niching, where the use of nearest-neighbour and clustering style techniques can yield $O(n^2)$ processing times [13] (as in SPEA and SPEAII). Moreover, in the general case, no optimisation of these niching procedures exists [13] and thus alternatives must be sought.

Beyond run-time efficiency, the performance of traditional multiobjective optimisers (such as MOGA, NSGA and NPGA) is tightly bound and extremely sensitive to the bias assigned to diversity preservation [1]. A failure to correctly specify the weighting of diversity in fitness assignment (typically referred to as the sharing factor) can lead to systems that prematurely diverge or converge. Thus, while guidelines exist for approximating appropriate sharing factors (see [4]), most practical systems will require significant tuning of this parameter to achieve optimal results.

Partially to address the inherent complexities associated with existing techniques, NSGAII introduces a more cost-effective approach to diversity maintenance that avoids $O(n^2)$ processing time and excessive parameter tuning. By utilising a simple crowding-distance metric, which exploits $O(\log n)$ objective-value sorting to enable low-cost nearest-neighbour measurements, NSGAII reduces diversity preservation complexity to $O(n \log n)$. Moreover, since the crowding-distance is only considered when breaking fitness-ties during tournament selection, no fitness sharing parameter is required. While the improvements made over existing techniques are impressive, complexity remains non-linear and evidence suggests ([1] citing [7]) that a notable search bias inhibits performance on higher objective problems.

In contrast, both PAES and PESA employ diversity-maintenance strategies that incur only linear complexity. These approaches divide the objective space into a hyper-grid and use the number of solutions occupying each cell to determine the relative crowding of that area. Although the move towards linear complexity is an important practical improvement, both approaches require the definition of cell-sizes, which will inevitably lead to additional parameter tuning. Furthermore, because the nature of grids is coarse, there is potential for the approach to miss or de-emphasise narrow regions of unexplored space.

Irrespective of chosen approach though, diversity preservation remains a complex, and largely unsolved, problem. Although existing techniques effectively distribute solutions across the objective-space, virtually all lead to optimisers that are susceptible to front deterioration due to the successive replacement of non-dominated solutions [17]. Such inability to maintain important solutions is significant and illustrates the complexities associated with balancing MOEA design – overly elitist approaches will lack the diversity to derive a successful spread along the Pareto optimal front, while diversity-preservation can slow and even prevent convergence onto the front. The negotiation of such issues, in addition to a continued focus on technique development, is of significant importance and warrants further attention in the MOEA research community.

2.3 Archive-Based Elitism

The stochastic nature of multiobjective optimisers – where the final set of solutions may not be representative of the best set of solutions found – requires that most, if not all ([18] citing [19]), practical installations of MOEAs capitalise on some form of solution repository. While traditional techniques use archiving purely as a background storage device, more contemporary approaches have included archival solutions as part of the selection process (SPEA, SPEAII and PAES) or as active members of the core population (PESA and NSGAII). By incorporating the archive into the evolutionary process, contemporary algorithms employ explicit elitism to bias the search around areas that have previously yielded the best results.

While empirical outcomes suggest that the incorporation of elitism into existing multiobjective algorithms can yield significant benefits [12], the use of archiving is not without limitations. Since the archive is now an active participant in the evolutionary process, careful bounds must be placed on its size to limit the negative impact that population growth will have on run-time complexity. Furthermore, since archives are typically composed of non-dominated solutions, archive maintenance can be complex and is largely based on diversity preservation principles – which, as seen earlier (Section 2.2), are both difficult to balance and potentially costly to execute. Moreover, as with any elitist approach, active-archiving infers a marked increase in selection pressure around promising solutions that can potentially lead to stagnation and premature convergence [1]. Thus, as with most core-concepts in MOEA design, archive-based elitism is as affected by complexity and balance limitations, as it is effective in achieving more rapid optimal convergence.

3 The Combative Accretion Model

The observations made in previous sections are not designed to cast popular pre-existing methods under a negative light, but to illustrate that while core MOEA techniques have achieved impressive results, they are not without their flaws. With this in mind, the development of unique approaches that aim to address existing problems can only aid in the continuing refinement and growth of multiobjective optimisers as a whole. It is such motivation that has led to the burgeoning growth of multiobjective research in areas as diverse as Artificial Life, Ant Colony simulation, Simulated Annealing and Messy Genetic Algorithms. It is also the primary motivation for this work: to develop a disparate, novel approach to multiobjective optimisation that explores new avenues for MOEA design while addressing the problems inherent in existing approaches.

In particular, the Combative Accretion Model is focussed on the reduction of complexity through the incorporation of implicit ranking, the removal of expensive diversity measures and a departure from conventional elitist archive design. Moreover, this goal is achieved in a unique system that is grounded in pair-wise dominance-based confrontation and accretion agent generation.

3.1 Agent Interaction

Central to CAM is the notion of agents and agent interaction. Borrowing terminology from the Artificial Life community, an agent is representative of a complete solution to the given multiobjective problem and carries an explicit mutable size that is representative of performance in the population[2].

Agent interaction is strictly pair-wise and the results are dictated by the dominance relationship between the two individuals (see Section 3.5). Thus, combat in CAM is derivative of binary tournaments, though selection probabilities are based around agent size rather than an expensive Pareto rank. Furthermore, the result of an interaction does not necessarily infer agent reproduction, as in conventional tournaments, but rather dictates agent survival and changes in agent size (see Section 3.5).

Since the size of an agent is the basis for selection and is subject to performance-based change, it represents endogenous fitness and results in an implicit and adaptable ranking of importance within the population – the greater the size of an agent, the more influence it will have. Such ordering is particularly significant when considering the efficiency of locating dominated solutions. In a traditional Pareto-ranking scheme, dominance is determined through an inherently expensive linear search of the current population – it does not fully capitalise on the performance of previously ranked solutions. Contrastingly, in CAM, a solution will be biased towards comparisons against the more successful agents – facilitating more rapid determination of dominated solutions (since these will generally perform poorly against the large, non-dominated agents).

3.2 The Gene Pool

To drive agents towards the current Pareto front, CAM makes use of a unique elitist concept based around the temporary storage of genes sourced from successful agents. This gene accumulation, which is ultimately used in agent creation (see Section 3.3), is referred to as the gene pool and is a finitely sized collection of alleles for each gene position. The pool is updated by any agent that passes a pre-specified size threshold (and is thus considered suitably fit) – with a random member of each gene-position collection (GPC) replaced by the corresponding agent allele (as outlined in the following equation and in Figure 1):

$$\forall i \in \{0,1,..,g-1\},\ GPC_{i_{\lfloor R|A| \rfloor}} = x_i \qquad \textbf{(1)}$$

where g is the number of genes per solution; x is the collection of agent genes; R is a random number between 0 and 1; GPC_i is the gene-position collection for the i^{th} gene position; and A is the array of alleles in the given GPC.

It is tempting to find a correlation between the gene pool concept and the notion of building blocks seen in messy genetic algorithms (see [20, 21]) – however, the two approaches are quite significantly different. Messy Genetic Algorithms are charged

---

[2] Solution, agent and chromosome are considered synonymous in this work. Alleles and genes will be used to refer to specific components of a solution.

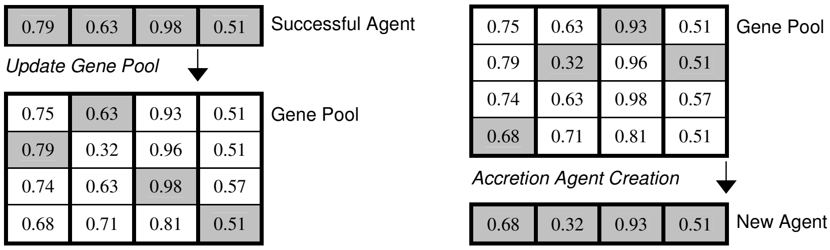


Fig. 1. Using the Gene Pool

with establishing the best possible linkage of building blocks in order to generate good solutions. In contrast, the gene pool concept utilises good solutions to identify important genes. Thus, the two approaches both aim to build on apparently successful components to drive evolution towards the Pareto optimal front, but the way in which those components are identified is diametrically opposed.

3.3 Recombination

Though reproduction can be of an asexual nature, most agent generation is performed via accretion creation – whereby the gene pool is randomly harvested to create a new individual (see Figure 1 & Equation 2). Specifically, for each GPC in the gene pool, a random member is selected and included as part of the new chromosome (with potential mutation dictated by mutation rate μ, and the new agent represented by x):

$$\forall i \in \{0,1,..,g-1\},\ x_i = GPC_{i_{|R|A||}};\ \text{if}\ (R < \mu)\ \text{then mutate}(x_i) \quad (2)$$

To ensure a static population size, agent creation only ever occurs upon the death of another agent. Thus, the system is essentially a continuous poor-performer replacement scheme, whereby successful agents are retained simply by surviving.

3.4 Agent Death and Elitism

In CAM, agent death occurs under special conditions of domination or when an agent passes a pre-determined exhaustion threshold (see Section 3.5). The exhaustion threshold, which dictates the maximum number of times a single agent can contribute to the gene pool, in association with the maximum size threshold, determines the level of system elitism. Increasing the size and exhaustion thresholds infers greater pressure on high performance, while decreases improve the likelihood of diversity in the system by reducing the influence of dominant agents. The relationship between these two parameters (in addition to gene pool size) and the development of corresponding heuristics and automated tuning techniques are important areas of future work that will maximise the simplicity and practical applicability of CAM.

3.5 The Algorithm

Figure 2 illustrates a typical execution of the CAM system. Note that unlike conventional approaches to MOEA design, the system is non-discrete (with respect to generations) and follows the more continuous approach adopted in contemporary Artificial Life systems (see [16]). Such a departure facilitates the exclusion of an explicit active-archive, as the population will almost always be composed of a combination of recently generated solutions and previously successful agents. Furthermore, note that beyond the influence of thresholds and simple checks for equality[3] and incomparability, there is no explicit diversity operator charged with keeping a well-spread distribution of solutions. Results will indicate the effect of excluding such expensive techniques under the current CAM implementation, though the base model itself does not preclude their use.

Also significant is the initialisation procedure – where both the agent set and gene pool are randomly filled. While the effect is likely to be minimal and should only impede early recombinations, it is perhaps preferable to initialise the gene pool to the empty set and prevent agent generation until the pool is at least partially filled with

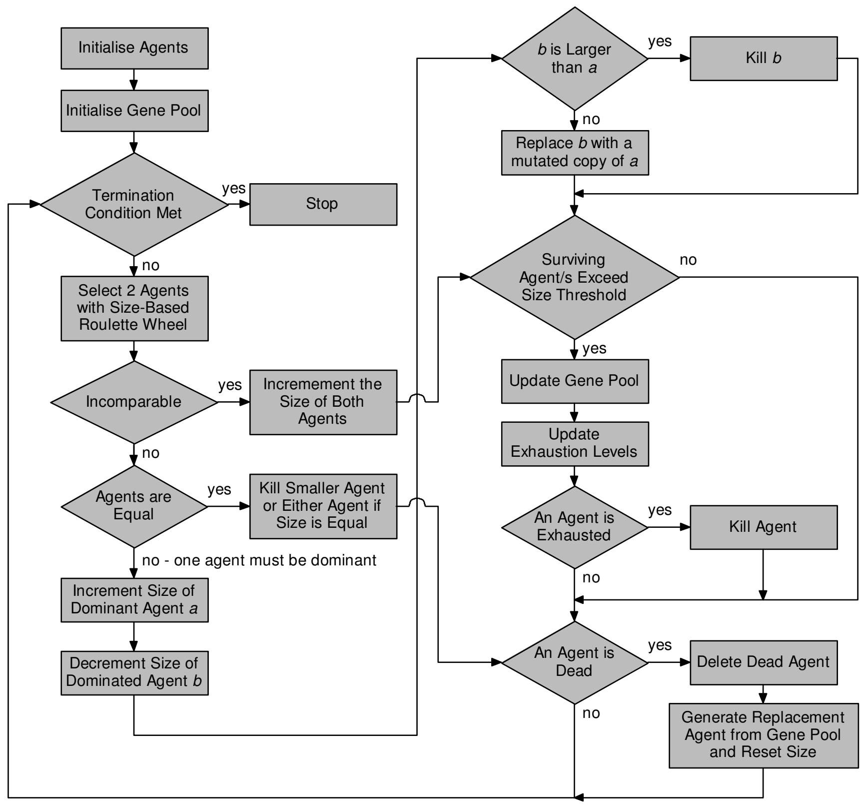


Fig. 2. The Execution Cycle of a CAM System

3 Currently, agent equality is assessed in objective-space, though this is not a requirement of the system.

potentially successful genes. Future work should consider this trivial amendment, though there does exist some potential for over-elitism if the pool is particularly sparse across a significant number of accretion creations.

Of the remaining processes, most are self-explanatory or have been discussed in previous sections, though both selection and combat may require further elucidation. The selection of agents for combat is as-per conventional binary roulette-wheel selection, though traditional fitness measures are replaced by a simple normalisation of agent size (using the corresponding size threshold). In the case of combat, to maximise diversity of the population, equality always leads to the death of one agent and incomparability results in growth of both agents. When an agent is dominated in combat it always dies, being replaced by a mutated clone of the dominant agent (when the size of the dominating agent is large and is thus likely to constitute a good solution) or an accretion creation (when the size of the dominating agent is small).

4 Results

Within the Multiobjective research community much debate exists as to which of a diverse set of performance metrics provide the most accurate representation of optimiser performance (see [22, 23] for a sample). This paper does not seek to settle the debate, but uses a broad range of both complexity and front analysis metrics that provide a detailed picture of CAM efficiency and effectiveness (see Table 1). To further elucidate CAM performance, and place it in a contemporary context, results are compared with both NSGA and NSGAII. The choice of systems here is important: both are popular techniques, with the original NSGA providing comparison with a non-elitist approach and NSGAII illustrating performance against an elitist system that has yielded impressive results [12] and is explicitly charged with reducing complexity.

All systems are tested on a broad range of well-recognised problems (see [12] for details) that emphasise the characteristics typically found in real world multiobjective optimisation – namely: convex (T1), concave (T2) and discontinuous fronts (T3); multi-modality (T4); and non-uniformly distributed fronts (T6). Note that since this CAM implementation is designed for real-value use only, T5 is excluded. The extension of CAM into binary problem domains lies as an important area of future work.

The presented results are representative of runs using test parameters specified in Table 2, with duplicate objective-space values removed to negate unreasonable biasing of the distribution metric. Parameter settings for NSGA and NSGAII are derived from system-defined defaults included in pre-existing implementations, or otherwise according to [5].

CAM parameter values underwent only limited tuning and thus further refinement and the associated development of corresponding heuristics can only improve overall performance. Note also that mutation in CAM is non-gaussian[4] – this is largely an

---

[4] Strictly, a distribution (with probabilities of 0.25, 0.375 and 0.375 respectively) between random ($v' = R(max\text{-}min)+min$), geometric ($v' = v \pm 0.075v$) and incremental ($v' = v \pm 0.075(max\text{-}min)$) mutation, where v is the initial value; v' is the new value; and *max* and *min* represent the range of allowed values.

arbitrary choice, though it is inspired by its recent use in Artificial Life systems. It may be beneficial to investigate the relative utility of this choice over more conventional operators in subsequent studies.

Table 1. Description of metrics used for system analysis

Metric	Motivation	Methodology
Front Graph	Establishes a visual hierarchy between competing systems and illustrates the proximity of produced solutions to the Pareto optimal front.	For each test display all non-dominated points from the amalgamation of solutions produced across three runs.
Coverage [6]	Provides a relative comparison between two non-dominated sets – determining, for one system, the ratio of solutions that are non-dominated with respect to a front produced by another system. Specifies which front is preferable – but not by how much.	For each test and system, compare each of the non-dominated sets produced across fifty runs against those sets produced by a competitor. Display the average.
Avg. Distance [12]	Specifies the average distance from a front produced by a given system to the Pareto optimal front. Thus, this measure encapsulates the accuracy of a system.	For each test and system, calculate the average Euclidean distance from the non-dominated solutions found in a given run to the Pareto optimal front. Average across fifty runs.
Extent [12]	Describes how widely spread solutions are by measuring the Euclidean distance between extreme points in a given front. The metric thus illustrates a system's ability to find boundary solutions.	For each test and system, calculate the average objective-space extent across the fifty non-dominated sets produced.
Delta Distribution [5]	Illustrates how well spread a front is by using the distance between consecutive solutions (d_i) to give: $\Delta = 1/n\left(\sum_{i=1}^{n}\lvert d_i - \bar{d}\rvert\right)$; where $\bar{d} = 1/n\left(\sum_{i=1}^{n} d_i\right)$ Note the large influence of front size in this function.	For each test and system, calculate the average objective-space distribution across the fifty non-dominated sets produced.
Population	Specifies the number of non-dominated solutions found across a given run. In combination with distribution, the population dictates the richness of the solution set.	For each test and system, find the number of unique objective-space solutions produced in each of the fifty runs. Display the average.
Complexity Graph	Establishes a visual hierarchy between competing systems by illustrating the average number of comparisons required per evaluation. A single comparison is considered to be any form of dominance check between two solutions.	For each test display the comparison averages per evaluation sourced from fifty distinct runs.
Big O Notation	Provides a theoretical assessment of the computational complexity of a given system.	Deduce the run-time complexity for the provided systems.

Table 2. System Settings

NSGAII and NSGA		CAM	
Uniform Crossover Rate	0.9	Gene Pool Collection Size	n/10
Gaussian Mutation Rate	1/g	Max Agent Size (s)	10
Crossover/Mutation Distribution Index	20	Initial/Reset Agent Size	$\lfloor s/2 \rfloor$
NSGA		Exhaustion Threshold (e)	30
Param. Space Sigma Share	$0.5 \cdot 0.1^{1/g}$	Non-Gaussian Mutation Rate	1/g
Common			
Population Size (n)	50	Termination Condition	Number of Evals.

4.1 Front Quality

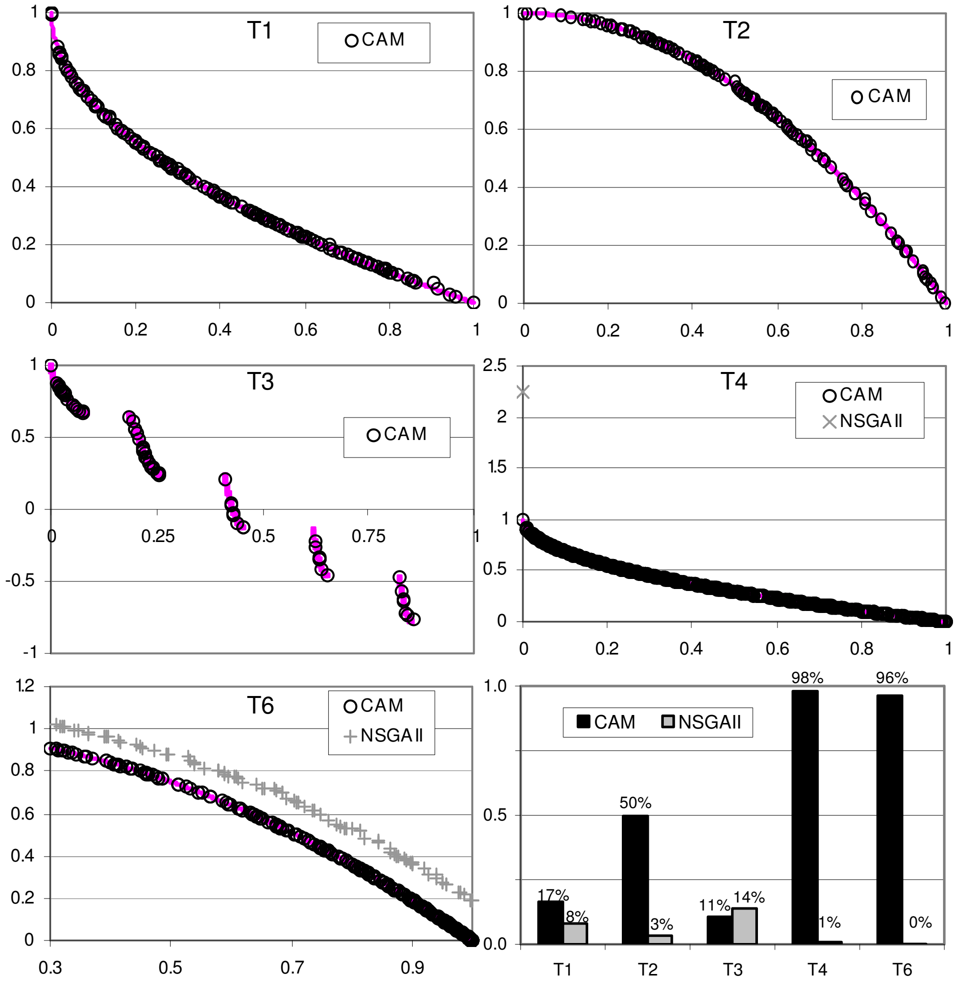
Figures 3-7 illustrate that for a broad range of problems, CAM is capable of finding both highly accurate and well-distributed approximations of the true Pareto optimal front. Furthermore, the arrival and development of these fronts is particularly efficient, requiring at most 5000 evaluations for all but the complex and multi-modal T4 problem. While there is degradation in overall efficiency for CAM in this instance, it still displays a marked improvement over NSGAII, which fails to converge to the true Pareto optimal front and stagnates in local optima. Moreover, T4 is generally recognised as the most difficult problem in the test-suite and existing studies have shown that systems such as SPEA and NPGA fail to locate the Pareto front even with larger populations [12]. Thus, results indicate that CAM can perform well on complex problem domains and is capable of moving through false local fronts with only minimal population sizes.

Furthermore, CAM clearly outperforms NSGAII on T6 – a non-uniform problem that has again been shown to cause difficulties for existing techniques such as SPEA [12]. Such robustness of performance irrespective of domain characteristics is an important feature of the CAM system and suggests broad practical applicability.

In addition to the superiority shown in the T4 and T6 graphs, coverage measurements (Figure 8) illustrate that CAM fronts are preferable to those produced by NSGAII on all remaining problems (excluding T3, where the systems are approximately equivalent). Moreover, while not displayed, CAM fronts completely dominate those generated by NSGA on every test function excluding T6 (where CAM achieves 94% coverage). Given that NSGA is a popular early system that forgoes archiving, such a comprehensive improvement by CAM is particularly significant.

To further clarify the contributing factors that define a given front, Figure 9 illustrates the average Euclidean distance, extent, distribution and non-dominated set size for each of the systems across the given test functions. In all cases, CAM has improved accuracy compared to both NSGA and NSGAII, finding solutions that are more closely positioned to the Pareto optimal front. Such a feature, in league with good extent values, illustrates the ability of CAM to rapidly develop highly accurate solutions, without converging onto a small region of objective-space. Indeed, across the entirety of the five fifty-run tests, the resultant CAM fronts never converged onto a single non-dominated point. Such avoidance of solution homogeny is particularly significant given the propensity for NSGA and NSGAII to become fixated on narrow areas of the objective-space (as in T4, where the resultant fronts of NSGAII and NSGA converged to a single non-dominated solution in 28% and 92% of runs respectively; and in T2, where NSGAII had such convergence in 42% of the runs).

However, it is important to note that while CAM features acceptable distribution levels which exceed those of NSGA (excluding T3), it is comparatively worse than NSGAII (excluding T6 and single-member populations). Such a difference is likely caused, and at least heavily influenced, by the increased frontal occupation of NSGAII (when it avoids single-point convergence), which reflects a more richly populated, though lower quality, approximation set than CAM. Thus, future work should examine the use of diversity-guided reproduction (via the manipulation of gene pool mechanics) to aid in the development of higher-cardinality high-quality sets in CAM.



	T1			T2*			T3		
	NSGAII	CAM	NSGA	NSGAII	CAM	NSGA	NSGAII	CAM	NSGA
Avg. Dist	0.0028(0.00)	0.0014(0.00)	2.2671(0.17)	0.0018(0.00)	0.0010(0.00)	3.0393(0.21)	0.0011(0.00)	0.0008(0.00)	2.0962(0.19)
Extent	1.3191(0.16)	1.4064(0.03)	2.4880(0.42)	0.7850(0.69)	1.4128(0.01)	0.6893(0.31)	1.8165(0.27)	1.9326(0.04)	2.9583(0.43)
Distribution	0.0038(0.00)	0.0174(0.00)	0.0267(0.01)	∞|0.0046(0.0)	0.0219(0.00)	0.0512(0.04)	0.0089(0.00)	0.0588(0.01)	0.0297(0.01)
Population	221.44(33.3)	56.76(6.7)	87.62(12.6)	116.36(102.2)	46.92(5.8)	16.62(7.3)	192.56(29.2)	29.40(4.3)	91.08(11.6)

	T4*				T6		
Avg. Dist	4.3597(3.65)	0.0558(0.10)	18.7708(8.72)	*Note that single member populations infer a distribution value of ∞. In this case, an alternative average is included which reflects the distribution of the remaining, better populated, sets	0.1066(0.01)	0.0067(0.01)	6.0263(0.28)
Extent	1.9163(1.35)	1.4436(0.06)	0.2138(0.97)		1.1347(0.03)	2.1787(2.02)	1.0434(0.32)
Distribution	∞|0.0008(0.0)	0.0122(0.01)	∞|0.2260(0.3)		0.0097(0.00)	0.0203(0.02)	0.1138(0.08)
Population	2028.02(1912)	149.76(77.0)	1.26(1.0)		73.3(10.8)	160.96(12.9)	13.52(3.9)

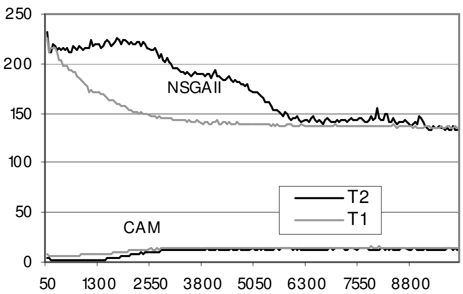
Figs. 3-9. Offline front analysis for runs of 5000 (T1-T3, T6) and 40000 evaluations (T4). **Figs. 3-7.** Non-dominated solutions produced during three distinct runs – NSGAII excluded when clarity of graph is lost; *reference line* indicates the optimal front; *x-axis* is objective one; *y*-axis is objective two. **Fig. 8.** The average relative coverage of non-dominated sets across fifty distinct runs. **Fig. 9.** Average performances on given metrics (standard deviations provided in brackets)

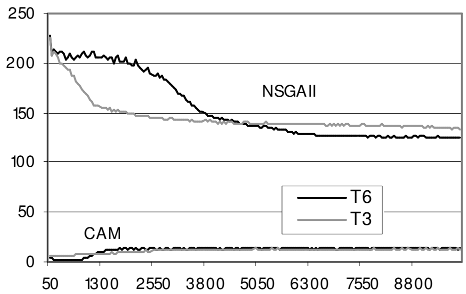
4.2 Complexity Analysis

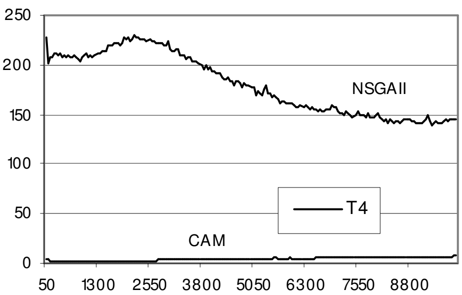
While the quality of fronts produced by CAM is impressive, the performance of any system cannot stand on quality alone. Indeed, for contemporary multiobjective optimisation, the utility of an algorithm is also contingent on the corresponding run-time complexity. With this in mind, the following section addresses the issue of complexity, utilising NSGAII to highlight relative performance improvements.

Empirical evidence illustrates that CAM is consistently and significantly more efficient than NSGAII across all of the tested areas for a population size of fifty (Figures 10-13). The poor performance of NSGAII can be attributed to the increased computational burden accrued from archiving, which essentially doubles the population size, sorting for fitness sharing and an explicit ranking scheme. By avoiding these techniques, CAM achieves an average run-time complexity that is faster than NSGAII by an order of magnitude and tends towards *O(ns)*.

The shape of the complexity graphs is also significant and warrants some discussion. CAM is most efficient early in the run where it can quickly identify dominated solutions via implicit size ordering (as discussed in Section 3.1). In sharp contrast, NSGAII is generally extremely inefficient during this period, requiring numerous passes through the population to determine explicit ranking, whilst lacking any existing Pareto ordering of solutions to maximise efficiency. Furthermore, the increase in the number of comparisons towards the completion of a CAM run indicates that dominance determination is flexible and adaptive, unlike the complexity reduction methods employed by NPGA. As the number of good solutions grow, so too does the breadth of search to ensure accurate non-dominated front representation







	Avg. Tot. Comparisons		Avg. Comps/Eval	
	NSGAII	CAM	NSGAII	CAM
T1	1,473,027	127,551	147.3	12.8
T2	1,738,467	105,946	173.8	10.6
T3	1,454,811	115,337	145.5	11.5
T4	1,795,247	43,197	179.5	4.3
T6	1,537,239	121,551	153.7	12.2

Figs. 10-13. Complexity Analysis. **Figs. 10-12.** Average number of comparisons (*y-axis*) per evaluation (*x-axis*) over fifty runs. **Fig. 13.** Summaries of the total number of comparisons and the overall average number of comparisons per evaluation across fifty runs of 10000 evaluations

– a capability that is beyond statically defined sub-population methods.

In terms of complexity, CAM performs worst when the population is highly non-dominated and incomparable – where solutions must generally wait until exhaustion to be removed. In this case, CAM has *O(nes)* performance. Since complexity becomes $O(n^2)$ when $es \cong n$, it is theoretically possible to have a worst case complexity equivalent to NSGAII. Since such complexity occurs only on near-complete and persistent stagnation of a well distributed optimal or pseudo-optimal front, it is unlikely for such complexity to occur prior to completion of a typical run (as evidenced in Figures 10-13). Moreover, since the complexity rise is directly tied to stagnation of well distributed fronts, at worst the performance of CAM degenerates to NSGAII levels when on potentially good fronts, and at best, this peak in complexity can be used as an additional termination condition (since it indicates either the generation of a good Pareto front approximation or premature front convergence).

5 Conclusions and Future Work

This paper has presented a novel approach to multiobjective optimisation that is driven by agent-based pair-wise dominance interactions and the development of an elitist gene pool for accretion agent creation. By avoiding explicit ranking, complex diversity preservation and expensive active-archiving procedures, results have shown that the Combative Accretion Model demonstrates an order-of-magnitude improvement over the run-time complexity of NSGAII across a wide range of functions. Moreover, CAM consistently produces good approximations of the Pareto optimal front irrespective of diverse problem characteristics, while achieving frontal quality that is typically beyond both NSGA and NSGAII – notions which are substantiated both graphically and through coverage and accuracy metrics. Given the promising nature of the achieved results, future work is certainly merited and should focus on extensions to the CAM system, the refinement and potential automation of parameters, and the application of the approach to real-world problems. In particular, work regarding the integration of diversity into the accretion process, or the use of a PAES-like hyper-grid for biasing pair-wise results, may aid in achieving more richly populated approximation sets.

Acknowledgements

The authors would like to acknowledge the following people and institutions, without whom this work would not have been possible: Trixie Berry, Michael Berry, Pauline Mak, Ian Lewis, David Benda, the School of Computing and the GAI research group.

References

1. Coello Coello, C.A., Lamont, G.B., and Van Veldhuizen, D.A., ‘Evolutionary Algorithms for Solving Multi-Objective Problems’. 2002, New York: Kluwer Academic.

2. Goldberg, D.E., 'Genetic Algorithms in Search, Optimization, and Machine Learning' 1989, Reading, Massachusetts: Addison-Wesley.
3. Srinivas, N. and Deb, K., *Multiobjective Optimization using Non-dominated Sorting in Genetic Algorithms*. Evolutionary Computation, 1994. 2(3): p. 221 - 248.
4. Fonesca, C.M. and Fleming, P.J. *Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization*. in Genetic Algorithms: Proceedings of the Fifth International Conference. 1993. San Mateo, CA.
5. Deb, K., Agrawal, S., Pratab, A., and Meyarivan, T., *A Fast Elitist Multi-Objective Genetic Algorithm for Multi-Objective Optimization: NSGA-II*. Proceedings of the Parallel Problem Solving from Nature Conference VI, 2000: p. 849-858.
6. Zitzler, E. and Thiele, L., *Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach*. Lecture Notes in Computer Science, 1999.
7. Zitzler, E., Laumanns, M., and Thiele, L., *SPEA2: Improving the Strength Pareto Evolutionary Algorithm For Multiobjective Optimization*. Evolutionary Methods for Design, Optimization and Control, 2002.
8. Knowles, J. and Corne, D. *The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Pareto Multiobjective Optimisation*. in Proceedings of the Congress on Evolutionary Computation. 1999. Washington, D.C.: IEEE Press.
9. Corne, D.W., Knowles, J.D., and Oates, M.J., *The Pareto-Envelope based Selection Algorithm for Multiobjective Optimisation*. PPSN VI, 2000: p. 869-878.
10. Corne, D.W., Jerram, N.R., Knowles, J.D., and Oates, M.J. *PESA-II: Region-Based Selection in Evolutionary Multiobjective Optimization*. in Proceedings of GECCO. 2001.
11. Horn, J., Nafpliotis, N., and Goldberg, D.E. *A Niched Pareto Genetic Algorithm for Multiobjective Optimization*. in Proceedings of the First CEC. 1994.
12. Zitzler, E., Thiele, L., and Deb, K. *Comparison of Multiobjective Evolutionary Algorithms: Empirical Results*. in 1999 Genetic and Evolutionary Computation Conference. 1999.
13. Jensen, M.T., *Reducing the Run-time Complexity of Multi-Objective EAs: The NSGA-II and Other Algorithms*. IEEE Transactions on Evolutionary Computation, 2003. 7(5).
14. Socha, K. and Kisiel-Dorohinicki, M. *Agent-Based Evolutionary Multiobjective Optimisation*. in Proceedings of CEC 2002 - Congress on Evolutionary Computation. 2002.
15. Laumanns, M., Rudolph, G., and Schwefel, H.-P. *A Spatial Predator-Prey Approach to Multi-Objective Optimization: a Preliminary Study*. in PPSN V. 1998.
16. Berry, A. and Vamplew, P. *A Simplified Artificial Life Model for Multiobjective Optimisation: A Preliminary Report*. in The Congress on Evolutionary Computation (CEC). 2003. Canberra, ACT.
17. Laumanns, M., Thiele, L., Deb, K., and Zitzler, E., *Combing Convergence and Diversity in Evolutionary Multi-Objective Optimization*. Evolutionary Computation, 2002. 10(3).
18. Knowles, J.D. and Corne, D.W., *Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors*. IEEE Transactions on Evolutionary Computation, 2003. 7(2).
19. Horn, J., *Multicriterion Decision Making*. Handbook of Evolutionary Computation, 1997.
20. Van Veldhuizen, D.A. and Lamont, G.B., *Multiobjective Optimization with Messy Genetic Algorithms*. Symposium on Applied Computing, 2000: p. 470-476.
21. Zydallis, J.B., Van Veldhuizen, D.A., and Lamont, G., *A Statistical Comparison of Multiobjective Evolutionary Algorithms Including the MOMGA-II*. EMO 2001, 2001.
22. Okabe, T., Jin, Y., and Sendhoff, B. *A Critical Survey of Performance Indices for Multi-Objective Optimisation*. in Congress on Evolutionary Computation. 2003. Canberra.
23. Zitzler, E., Thiele, L., Laumanns, M., Fonesca, C.M., and Grunert de Fonesca, V., *Performance Assessment of Multiobjective Optimizers: An Analysis and Review*. IEEE Transactions on Evolutionary Computation (Accepted for Publication), 2002.