Harnessing Phenotypic Diversity towards Multiple Independent Objectives

TOKARCHUK, LN; Smith, D; Wiggins, G; Genetic and Evolutionary Computation Conference

Harnessing Phenotypic Diversity towards Multiple Independent Objectives

Davy Smith, Laurissa Tokarchuk and Geraint Wiggins
Queen Mary University of London
School of Electronic Engineering and Computer Science
London, E1 4NS, UK
{david.smith,laurissa.tokarchuk,geraint.wiggins}@qmul.ac.uk

ABSTRACT

Multiple assessment directed novelty search (MADNS), introduced by the authors in [20], is an extension to the novelty search algorithm which exploits the observation that populations optimised for phenotypic novelty may contain solutions to multiple independent and conflicting objectives. It has been shown that, through the application of MADNS, an evolutionary trajectory may be simultaneously directed towards multiple conflicting objectives. Previous results from a series of simulated maze navigation experiments have shown that MADNS may significantly outperform novelty search in domains where the potential for phenotypic exploration is high [20]. In this paper we further explore the MADNS algorithm, assessing the effect upon the diversity and performance of the population as the phenotypic landscape increases. A series of experiments in domains with multiple conflicting objectives and expanding areas of *irrelevant space* show that the relative performance gain of MADNS increases alongside the potential for exploration. We conclude that, as the potential for exploration within a domain expands, the importance of directing novelty becomes ever more necessary.

CCS Concepts

•**Mathematics of computing** → **Evolutionary algorithms;** •**Computing methodologies** → **Evolutionary robotics;**

Keywords

Novelty search; algorithm design; phenotypic diversity; neuroevolution; evolutionary robotics

1. INTRODUCTION

Novelty search [9, 12, 18] is an evolutionary approach which promotes phenotypic diversity within a population. Novelty search is applied by replacing an objective fitness assessment with a behavioural distance metric, which is most commonly domain specific [9], however there have been studies which suggest the use of domain independent measures [7], or altering the metric during an evolutionary run [3]. Studies have also aimed to determine the effect that the individual heuristic parameters have on novelty search [5, 8].

Due to its exploratory nature, novelty search has been shown to outperform fitness based search, particularly in deceptive domains [9]. Extensions of novelty search have been applied to a range of tasks, including the evolution of programs [10], diverse sets of virtual creatures [13], procedural content generation in video games [14, 17] and training diverse video game playing agents [19]. However, to date, proposed novelty search variants have generally concentrated on novelty search as a diversity preservation mechanism, utilising the algorithm to produce diverse sets of solutions for a particular objective.

However, if the phenotypic state space is large, an evolutionary trajectory optimised through novelty search may focus upon areas of the landscape which are without use to the particular task. Without directing the search towards particular areas of interest, novelty search may be unable to provide solutions to the intended objective(s). Therefore, a method for directing novelty search could potentially allow the trajectory to focus upon areas of the landscape of interest to the particular task at hand.

Multiple assessment directed novelty search (MADNS) is an extension to the novelty search algorithm which utilises the diversity of potential solutions produced by optimisation through novelty [20]. MADNS exploits the observation that populations optimised through novelty search may contain solutions to multiple independent and conflicting objectives, and that the evolutionary trajectory may be optimised towards these objectives. Unlike multi-objective algorithms, which aim to optimise individual solutions to perform well over a number of objectives [2], and multi-modal algorithms, which aim to provide sets of solutions to objectives with multiple optima [4], MADNS utilises the diversity of a population optimised through novelty search to simultaneously provide individual solutions to multiple, potentially conflicting objectives. We believe that through directing populations of candidate solutions towards multiple objectives, the potential behavioural abilities of a genotypic representation becomes further uncovered — for example, simultaneously assessing the capacity of a robot controller to exhibit pursuit, evasion and foraging behaviours.

In this paper, we initially highlight the unsuitability of current fitness-based variants of novelty search to extend either beyond a single objective, or towards conflicting objectives. In section 3.1 we demonstrate how the linear combination of objective fitness is unsuitable for directing search towards multiple conflicting objectives.

In section 4 we further explore the MADNS algorithm, our method for directing novelty towards multiple objectives. We suggest variants of the MADNS algorithm based on both traditional novelty search and minimal criteria novelty search (MC-MADNS), highlighting the suitability to extend to unbounded domains

In section 5 we experimentally assess the relative effect

that increasing the potential for exploration has upon the performance and diversity of populations optimised through novelty search and MADNS. We present a series of maze navigation domains with multiple exits and differing levels of *irrelevant space*; namely, space within the maze which encourages phenotypic exploration and divergence from the objectives. Our results, presented in section 6, indicate that increasing the phenotypic landscape has significantly less effect upon the performance of MADNS compared to novelty search.

2. BACKGROUND

2.1 Novelty Search

Novelty search, as proposed by Lehman and Stanley [11], is an algorithm which removes the need for an objective fitness function through the assignment of high fitness values to novel behaviours in a population. The traditional objective fitness function is replaced by a behavioural distance metric, which is used to determine the novelty of an individual in a population. The behavioural distance metric measures phenotypic traits of the particular solution, and may be domain specific or a generalised, domain independent measurement.

$$f_{nov}(\rho) = \frac{1}{k} \sum_{i=0}^{k} dist(\rho, \mu_i) \qquad (1)$$

The behavioural novelty $f_{nov}(\rho)$ of an individual ρ, given by equation (1), is defined as the mean behavioural distance between ρ and its k nearest neighbours, where k is a user defined parameter and μ_i is the ith nearest neighbour of ρ with respect to the distance $dist$. The neighbours μ_i include both the behaviours of the current population and an archive of previous novel behaviours. Individuals with a value of $f_{nov}(\rho)$ above a predefined novelty threshold are added to the archive.

Novelty search receives increasing interest within evolutionary robotics research for its unique way in expanding the search space to multiple solutions in any given domain [9]. The introduction of novelty search uncovered many of the stepping stones towards open-ended evolution.

Although novelty search may outperform objective fitness search in specific tasks, especially when the design of an objective fitness function may be difficult, it has been shown that the assessment of behavioural novelty alone is insufficient as a generalisable evolutionary technique in many domains [1, 15]. Due to its divergent nature, novelty search continues to produce new solutions throughout the evolution, however these solutions may not be useful for the task at hand. Solutions which are sub-optimal for one particular objective may, however, be useful solutions to separate objectives. In this paper, we focus on the potential of the diversity of solutions provided by novelty search to be of use to a wide range of conflicting objectives.

2.2 Novelty Search Variants

Since the introduction of novelty search, various hybrid variants have been proposed to assist the phenotypic exploration potential of the algorithm towards particular objectives. Hybrid novelty search variants include minimal criteria novelty search [11], progressive minimal criteria novelty search [6], novelty search with local competition [13] and constrained novelty search [14]. Novelty search has also been combined with objective fitness search, both through linear mixing of the novelty and objective fitness values [19] and as separate objective in a multi objective algorithm, such as NSGA-II [15].

As minimal criteria novelty search (MCNS) was introduced in [11] to address the inability of novelty search to scale to domains with large phenotypic landscapes, this variant was selected as the most relevant for comparison in our experiments.

2.2.1 Minimal Criteria Novelty Search

MCNS was introduced to restrict the exploration potential of novelty search in domains with large phenotypic landscapes, [11]. Through restricting solutions which do not meet a predefined criteria, MCNS directs exploration towards *useful* areas of the phenotypic landscape. In domains where the potential for exploration is high, MCNS has been shown to outperform traditional novelty search [11]. MCNS is identical to novelty search, with the addition of a minimal criteria that each solution must meet, $f_{mc}(p) : P \mapsto \mathbb{B}$. Any solution which does not meet this criteria, is assigned a novelty score of 0.

$$f_{mcns}(p) = \begin{cases} f_{nov}(p) & \text{if } f_{mc}(p) \text{ is true} \\ 0 & \text{if } f_{mc}(p) \text{ is false} \end{cases} \qquad (2)$$

3. APPROACH

In this section we outline the approach taken to develop an algorithm capable of directing novelty search toward multiple conflicting objectives. We begin by highlighting the unsuitability of linear combination, due to the assessment criteria simultaneously assigning a high score to a particular agent which performs well at one objective and a low score to a conflicting objective.

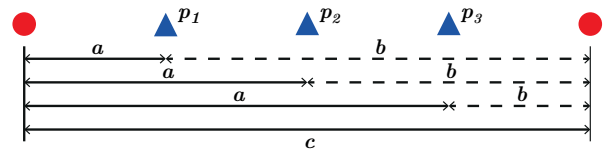### 3.1 Linear Combination of Objective Values

Figure 1: 1-dimensional domain

Here we demonstrate that the linear combination method is unsuitable for the directed optimisation of novelty search towards multiple conflicting objectives. Figure 1 illustrates a simple 1-dimensional domain with 2 directly conflicting objective destinations (figure 1, circles). The fitness of a particular solution, $f_k(\rho)$, is calculated through the linear combination of the distances from the position of the solution (figure 1, triangles, ρ) to each destination. The fitnesses to each objective are defined as: $f_1(\rho) = {(c-a)}/{c}$, $f_2(\rho) = {(c-b)}/{c}$. The overall fitness, $f_{lin}(\rho)$, is calculated as the linear combination of the objectives, ${f_1(\rho)}/{2} + {f_2(\rho)}/{2}$.

Through substitution, $f_{lin} = {}^{(2(a+b)-a-b)}/{(2(a+b))}$. Therefore, for all ρ, $f_{lin}(\rho) = {}^{1}/{2}$. Due to the fitness assignments all collapsing to the same value, optimisation would be unable to progress.

In agreement with our reasoning, the linear combination method is shown to be unsuccessful in our experiments (OBJ-LIN in section 5).

3.2 Maximum Objective Value

As highlighted above, the linear combination of fitness values is unable to extend towards multiple conflicting objectives. However, assigning the solutions' maximum fitness value for all of the objectives may allow the population to be optimised towards multiple conflicting objectives. This method, which can be linearly combined with novelty search, are defined as $f_{max}(\rho)$ and $f_{ns-max}(\rho)$, equation (3):

$$
\begin{aligned}
f_{max}(\rho) &= \max_{1 \le i \le k} \{f_i(\rho)\} \\
f_{ns-max}(\rho) &= \frac{f_{nov}(\rho)}{2} + \frac{f_{max}(\rho)}{2}
\end{aligned}
\tag{3}
$$

Both $f_{max}(\rho)$ (OBJ-MAX) and $f_{ns-max}(\rho)$ (NS-MAX) are assessed in section 5.

4. MULTIPLE ASSESSMENT DIRECTED NOVELTY SEARCH

MADNS [20] is an extension to the novelty search algorithm that rewards high performing solutions over a number of predefined objectives. MADNS is identical to novelty search, with the addition of a set of *assessments* — traditional objective fitness functions. Unlike hybrid novelty-objective variants which utilise Pareto efficiency or linear combination, the mechanism for fitness assignment in MADNS does not involve comparison between any of the assessments, which allows the functions to be partially or directly conflicting without negative effect on any particular assessment. Specifically, the MADNS fitness assignment, defined in equation (4), retains the novelty score for all members of the population other than the current highest ranking solutions for each of the assessments. These highest ranking solutions are assigned a score equal to the most novel solution in the current population.

Formally, let the set P denote the current population, with an individual solution defined as $\rho \in P$. Next, for a given domain with k objectives, defined by the functions $a_1(\rho), a_2(\rho), \ldots, a_k(\rho)$, where $a_k : P \mapsto \mathbb{R}$, let the set $A = \{a_1(\rho), a_2(\rho), \ldots, a_k(\rho)\}$. Let the subset $Q \subset P$ contain the maximal solutions for each objective, where:

$$
Q = \arg\max_{\rho \in P} a_1(\rho) \bigcup \arg\max_{\rho \in P} a_2(\rho) \bigcup ... \bigcup \arg\max_{\rho \in P} a_k(\rho).
$$

Let $f_{nov}(p)$, as defined in equation (1), be the novelty of a solution and the maximal novelty value for the current population be defined as $\alpha = \max_{\rho \in P} f_{nov}(\rho)$. Finally, let $f_{mad}(\rho)$ be the fitness of an individual solution, calculated as in equation (4):

$$
f_{mad}(p) = \begin{cases} f_{nov}(p) & \text{if } p \notin Q, \\ \alpha & \text{if } p \in Q. \end{cases}
\tag{4}
$$

A minimal criteria variant of this (MC-MADNS) may be defined through the replacement of novelty search with MCNS (section 2.2.1):

$$
f_{mc-mad}(p) = \begin{cases} f_{mcns}(p) & \text{if } p \notin Q, \\ \alpha & \text{if } p \in Q. \end{cases}
\tag{5}
$$

5. EXPERIMENT

5.1 Domain

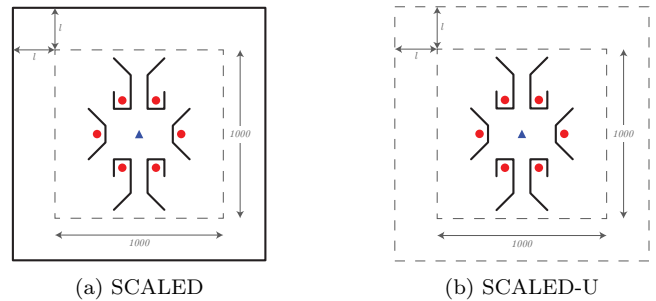
(a) SCALED   (b) SCALED-U

Figure 2: Maze domains tested in experiment.

Our experimental domain is based upon previous studies which have assessed novelty search and variants of the algorithm [9, 12, 5]. The task domain is a simulated maze, in which an agent controller must navigate from an initial starting-point to one of a possible number of exit points within a fixed time limit.

In this paper, we further explore 2 of the domains previously presented in [20], the SCALED and SCALED-U domains:

(a) SCALED — A divergent maze with 6 exits. The distance between the exits and the edges of the maze (figure 2a, $w = 1000 + l$) is altered in each evolutionary trial. Triangle indicates start position. Circles represent exits. The values tested are $l = \{0, 4000, 9000\}$. Values were tested in 50 evolutionary trials for each algorithm

(b) SCALED-U — A varying scale maze identical to SCALED with unbounded exploration potential, (figure 2b). This unbounded domain is a much more difficult task for divergent evolution, as the potential space for exploration is potentially infinite

Each of the mazes range from 1000-10,000 units, dependant upon the trial. The agent has a size of 20 units and the exits require the agent to be within 20 units of them. Each agent is given 4000 time steps to complete the maze.

5.2 Agent Model

The agent controllers in the maze experiments are neural networks, evolved using the NEAT algorithm [21]. In a similar manner to previous maze navigation experiments [9, 12, 5], the agent's perception (figure 3) utilises 6 range-finding sensors, which return the distance to the nearest obstacle.
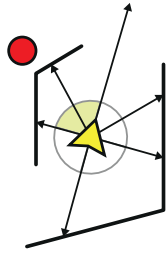
The range-finding sensors are mapped to $[0, 1]$, where 1 is equivalent to the diagonal distance of the full maze.

Figure 3: Agent vision

In addition to this, the agent has 4 inputs which act as pie-slice sensors. The pie-slice which contains the line between the agent and the exit is activated with a value of 1. For multiple exit mazes, the closest exit from the agent at any particular time step is used. The agent network has 4 outputs which control rotation left and right, and positive and negative thrust respectively.

5.3 Objective and Behavioural Definitions

The objective fitness function in maze domain tasks typically reward candidate solutions for a smaller euclidean distance between the ending position of the agent and the exit [9, 12, 5]. In line with previous studies, each of our objective fitness functions assesses the ending position of an agent to a particular exit. For the algorithms which compute a linear combination (NS-MAX, OBJ-LIN) all values must fall within the same range to avoid one value overpowering another. Therefore we scale the maximum distance of a maze with width w and height h to $\sqrt{w^2 + h^2} \mapsto 1$.

A single objective fitness of a particular agent ρ is given as $f_i(\rho) = 1 - dist(\rho, e_i)$, where e_i is a single exit. The fitness for each of the exits is calculated and assigned to each agent in a manner dependant upon the particular algorithm used. In the unbounded domains the end position of an agent may fall outside of the maze boundary, resulting in a distance greater than the defined size. To avoid negative fitness values, a constant is added to all solutions.

Again, in line with previous maze navigation tasks, the behavioural distance metric is calculated from the ending positions of the agent, $\beta = (\rho_x, \rho_y)$.

5.4 Evolutionary Criteria

In each evolutionary run, populations of 100 neural controllers were optimised for 1000 generations. The algorithms were run in each domain 50 times and were given a different random seed value for each run. In order to ensure consistency between algorithms, identical random seed values were given to each of the algorithms in each trial.

The simulation was performed using a bespoke domain written in the C++ programming language, developed to be similar to the original maze domain experiments in [9, 12]. The implementation of the NEAT algorithm used was developed as an extension to the MultiNEAT software in the C++ language [1]

5.5 Algorithms Tested

The following algorithms were tested in each of the maze domains:

(a) NS — Standard novelty search, as defined by $f_{nov}(\rho)$, (section 2.1 and equation (1)).

(b) MCNS — Minimal criteria novelty search, as defined by $f_{mcns}(\rho)$, (section 2.2.1 and equation (2)).

(c) MADNS — Multiple assessment directed novelty search, as defined by $f_{mad}(\rho)$, (section 4 and equation (4)).

(d) MC-MADNS — Multiple assessment directed minimal criteria novelty search, as defined by $f_{mc-mad}(\rho)$, (section 4 and equation (5)).

(e) OBJ-LIN — A linearly combined fitness of the distance to each of the exits in the domain, as defined by $f_{lin}(\rho)$, (section 3.1).

(f) OBJ-MAX — The maximum fitness value for each of the exits, i.e. the fitness to the closest exit at the end of the trial, as defined by $f_{max}(\rho)$, (section 3.2 and equation (3)).

(g) NS-MAX — A linear combination of NS and OBJ-MAX as defined by $f_{ns-max}(\rho)$, (section 3.2 and equation (3)).

5.6 Calculating Performance

The performance of each algorithm was determined by the probability of it to locate all solutions in the domain within a predetermined number of generations. If solutions to all of the exits were not found after 1000 generations, the trial was deemed unsuccessful. The probability of success for each algorithm in each of the domains was calculated by measuring the cumulative probability to discover all solutions to the exists within a single trial after n generations. figure 4 shows the probability of success for the algorithms in each of the domains tested. Comparisons of significant difference between two algorithms were calculated throughout using a two-tailed Mann-Whitney U test.

5.7 Calculating Population Diversity

In order to calculate the diversity of a particular algorithm, the domain is divided into a 2-dimensional matrix M, where $|M| = n \times n$. In our experiments, $n = 30$. The final position of an individual ρ, $pos(\rho) = (\rho_x, \rho_y)$, is mapped to the corresponding region of M. In a similar manner to [5], the *exploration uniformity* is calculated to determine diversity. However, we calculate diversity of the current population in each generation rather than the cumulative exploration uniformity over the whole evolutionary run. Let P_t be the set of individuals in the population at generation t and let Ψ_t be the distribution of P_t over M. The diversity of the population, $D(P_t)$, is calculated as the similarity between Ψ_t and the uniform distribution U. As in [5] the distance metric used is the *Jensen-Shannon* distance (JSD). The population diversity for a single generation is therefore
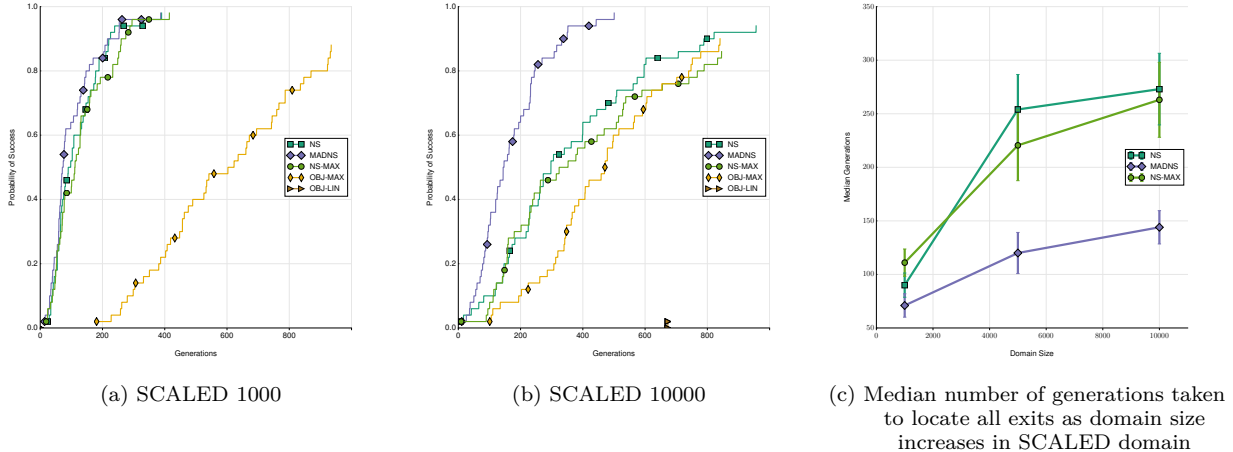
| (a) SCALED 1000 | (b) SCALED 10000 | (c) Median number of generations taken to locate all exits as domain size increases in SCALED domain |

Figure 4: Performance results for SCALED domain

defined as:

$$D(P_t) = 1 - JSD(\Psi_t, U), \ where:$$

$$\Psi_t = \left(\frac{|I_1|}{|P_t|}, ..., \frac{|I_{|P_t|}|}{|P_t|}\right), I_r = \{i \in P_t | region(i) = r\}$$

$$U = \left(\overbrace{\frac{1}{|M|} \times \cdots \times \frac{1}{|M|}}^{n^2 \ times}\right)$$

(6)

6. RESULTS

Here we present the probability of success (section 6.1, figure 4) as calculated in section 5.6 and the diversity (section 6.2, figure 5) as calculated in section 5.7 for each of the algorithms. In sections 6.1.1 and 6.2.1 we present the results for the bounded domain (SCALED). Our results for the unbounded domain (SCALED-U) are presented in sections 6.1.2 and 6.2.2.

6.1 Performance

6.1.1 Bounded Domain

As the SCALED maze is a bounded domain, the results for NS and MADNS are identical to their respective minimal criteria variants, MCNS and MC-MADNS.

Figure 4 shows the probability of success for each algorithm in the SCALED domain over 50 evolutionary runs, calculated as outlined in section 5.6. Figure 4a illustrates the probability of success for the smallest domain ($w = 1000$). There is no significant difference between NS and MADNS in this domain, with both algorithms locating all exits in all trials within 400 generations. MADNS slightly outperforms NS on both the maximum number of generations taken to locate all exits, (MADNS = 387, NS = 390) and the median value (MADNS = 71.0, NS = 90.0).

However, as shown in figure 4b, MADNS significantly outperforms NS in the SCALED domain when $w = 10000$. The

results for NS in the largest domain are similar in performance to OBJ-MAX. The difference in performance between MADNS and NS can be seen to increase as the domain size increases, as shown in figure 4c. NS-MAX performs similarly to NS, with no significant difference between the algorithms in any of the domain sizes (figures 4a to 4c).

6.1.2 Unbounded Domain

	number of exits located						
Algorithm	0	1	2	3	4	5	6
NS	25	20	2	2	1	0	0
MADNS	6	28	11	3	0	0	2
MCNS	2	25	17	3	2	1	0
MC-MADNS	0	5	25	9	6	2	3

Table 1: Frequencies of the number of exits located within 1000 generations. SCALED-U, $w = 1000$.

The probability of success is lower for the SCALED-U domain (probs) over the 1000 generations. In our previous experiment, presented in [20], many of the trials required over 1000 generations to locate all of the exits in the SCALED-U domain. We therefore expect that a longer evolutionary run would increase the number of trials which are successful in locating all exits. Tables 1 and 2 presents the number of exits the algorithms managed to locate within the allocated 1000 generations. In the smaller domain ($w = 1000$, table 1) only MADNS and MC-MADNS manage to locate all exits in any of the trials. MADNS significantly outperforms NS ($p \leq 0.01$ 2 tailed Mann-Whitney U) and MC-MADNS significantly outperforms MCNS ($p \leq 0.01$). MC-MADNS performs most optimally, locating at least 1 exit in all trials.

In the larger domain ($w = 10000$), all algorithms perform sub-optimally. However, NS and MCNS are more severely affected. Surprisingly, MCNS has the worst performance out of all of the algorithms, locating no exits in 41 out of 50 trials. Similarly, NS locates no exits in 34 out of 50 trials. As previously noted, is it expected that these results would improve with a longer evolutionary trail length. This may
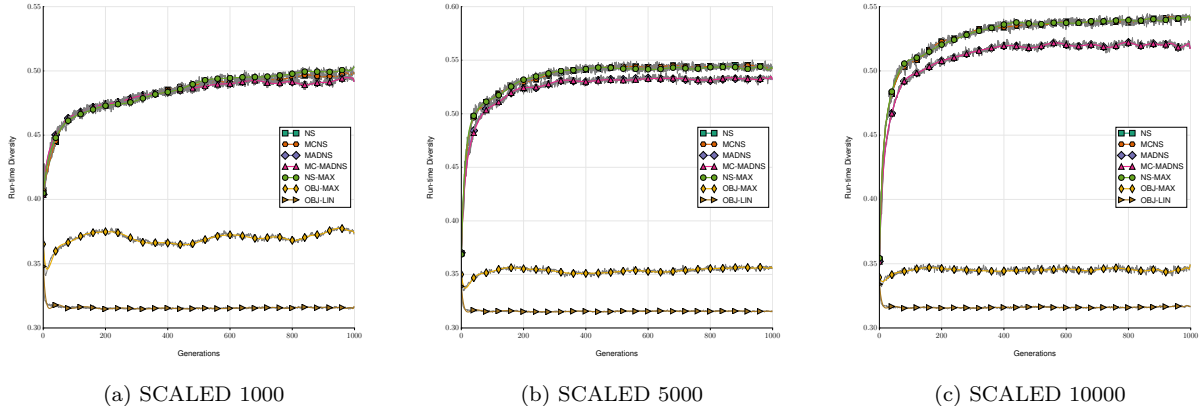
| (a) SCALED 1000 | (b) SCALED 5000 | (c) SCALED 10000 |

Figure 5: Uniformity exploration within the population at each generation (SCALED)

	number of exits located						
Algorithm	0	1	2	3	4	5	6
NS	34	12	4	0	0	0	0
MADNS	21	21	6	0	2	0	0
MCNS	41	4	3	2	0	0	0
MC-MADNS	20	18	10	2	0	0	0

Table 2: Frequencies of the number of exits located within 1000 generations in the unbounded SCALED-U domain ($w = 10000$).

be due to a large number of solutions being drawn outside of the bounds of the domain (figure 7d).

6.2 Diversity

6.2.1 Bounded Domain

Figure 5 shows the exploration uniformity for the SCALED domains for $w = 1000$ (figure 5a), $w = 5000$ (figure 5b) and $w = 10000$ (figure 5c). As expected, in a similar manner to section 6.1, results for the minimal criteria variants are identical for this domain, therefore have been excluded from discussion. In all cases, populations optimised through NS exhibit greater diversity in the population than MADNS. As in section 6.1.1, NS-MAX performs similarly to NS, with no significant difference between the algorithms in any of the domain sizes. The relative difference between the levels of exploration uniformity for NS and MADNS increases alongside domain size (figures 5a to 5c).

6.2.2 Unbounded Domain

The diversity is extremely unstable in the SCALED-U domain for all sizes of w (figure 6). However, due to solutions falling outside of the domain being penalised in the minimal criteria variants, the effect upon these algorithms is less severe. MCNS and MC-MADNS perform relatively similarly throughout all domain sizes. However, as the domain size increases, the benefit gained by the minimal criteria decreases (figures 6a and 6c). In a similar manner to performance, all algorithms still seem to be optimising at the end of the 1000 generations, therefore a longer trial period may be necessary in the unbounded domain.

7. DISCUSSION

Figure 7 illustrates a series of full evolutionary runs in the SCALED (figures 7a and 7b) and SCALED-U (figures 7c and 7d) domains. The figures indicate the final resting positions of every agent over the course of the full trial. As we can see, in the bounded domain both NS and MADNS explore the full area of the maze. However, in the unbounded domain, SUBSET-U, the divergent nature of NS pushes the search to focus upon areas of the landscape outside of the domain. MADNS resists this pressure for exploration (and high novelty), by constantly attracting the search towards the areas of interest.

The algorithms struggle to optimise all exits in the unbounded domain. We suggest that, in the bounded domain, the borders of the maze act as not only barriers, but also as reference points to the agent, aiding the controller to position itself in novel areas of the maze. Without this frame of reference and with the added potential for a controller to leave the edges of the maze in all directions, the divergent evolution struggles.

As we have shown in section 6, as the phenotypic landscape increases, the divergent evolutionary trajectory pursued by NS requires more assistance to focus upon areas of interest. This is hardly surprising, as NS purposefully has no concept of an objective built within the algorithm. In fact, NS was introduced with the specific desire to *abandon objectives* [12], alleviating the tendency for search to be drawn towards local optima, particularly in deceptive domains. This divergent exploration technique has proved successful in domains with small phenotypic landscapes, however, as the landscape expands, the search for pure novelty often requires assistance to focus the optimisation. While MCNS goes somewhat towards restricting the exploration, as we have shown, specifically directing the search through MC-MADNS assists further still, without fully removing the benefits of divergent exploration.

Our experiments also indicate that, potentially any area of the phenotypic landscape could be used as an attractor, regardless of the objective assigned to it. However, further studies in domains where the mapping between the phenotypic landscape and the objective function are less direct would be needed to fully establish the relationship between phenotypic exploration and objective functionality.
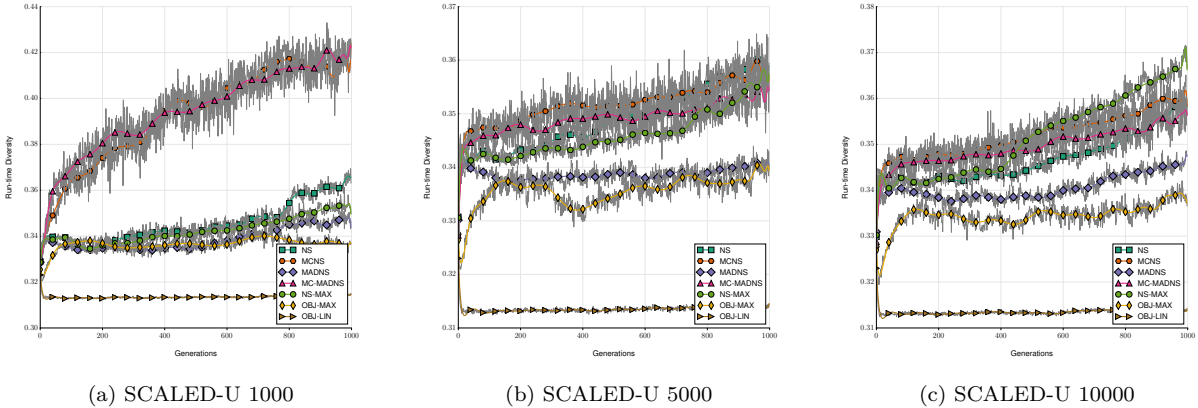
| (a) SCALED-U 1000 | (b) SCALED-U 5000 | (c) SCALED-U 10000 |

Figure 6: Uniformity exploration within the population at each generation (SCALED-U)

8. CONCLUSION

In this paper, we further explored the MADNS algorithm to ascertain whether increasing the potential for exploration would have a negative effect upon the performance of the algorithm. Our results have shown that, in domains which have a low exploration potential, NS is an effective method for the optimisation of populations towards multiple conflicting objectives. However, in domains which have high exploration potential, NS and MCNS are severely negatively effected. The MADNS and MC-MADNS algorithms have a higher probability of success than both NS and MCNS in such domains.

Our results also show that directing towards multiple objectives through our proposed method has little negative effect on the population diversity, rather that the diversity within the population is focused towards areas of the landscape of interest to the particular objectives. We have also shown that the particular objectives may be independent and directly conflicting, without any negative effect on the performance of the MADNS algorithm.

Previous studies have indicated that a high probability of success in maze navigation domains is generally transferable to more complex scenarios [5, 16]. Further testing of the MADNS algorithm in more complex domains may be necessary to conclude this.

Possible further scenarios which may be suitable for the algorithm's application include the evolution of predator-prey dynamics simultaneously in a single population, or the evolution of non-player-character video game agent populations which act as both enemies and companions to the player.

8.1 Acknowledgements

9. REFERENCES

[1] G. Cuccu and F. Gomez. When novelty is not enough. In *Applications of Evolutionary Computation*, pages 234–243. Springer, 2011.

[2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.

[3] S. Doncieux and J.-B. Mouret. Dynamic behavioral diversity. In *Proceedings of the 14th International Conference on Genetic and Evolutionary Computation Conference*, pages 1453–1454. ACM, 2012.

[4] M. G. Epitropakis, X. Li, and E. K. Burke. A dynamic archive niching differential evolution algorithm for multimodal optimization. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 79–86. IEEE, 2013.

[5] J. Gomes, P. Mariano, and A. L. Christensen. Devising effective novelty search algorithms: A comprehensive empirical study. In *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference*, pages 943–950. ACM, 2015.

[6] J. Gomes, P. Urbano, and A. L. Christensen. Progressive minimal criteria novelty search. In *Advances in Artificial Intelligence–IBERAMIA 2012*, pages 281–290. Springer, 2012.

[7] F. J. Gomez. Sustaining diversity using behavioral information distance. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 113–120. ACM, 2009.

[8] S. Kistemaker and S. Whiteson. Critical factors in the performance of novelty search. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 965–972. ACM, 2011.

[9] J. Lehman and K. O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pages 329–336, 2008.

[10] J. Lehman and K. O. Stanley. Efficiently evolving programs through the search for novelty. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, pages 837–844. ACM, 2010.

[11] J. Lehman and K. O. Stanley. Revising the evolutionary computation abstraction: minimal criteria novelty search. In *Proceedings of the 12th*

(a) MADNS in SCALED domain   (b) NS in SCALED domain   (c) MADNS in SCALED-U domain   (d) NS in SCALED-U domain
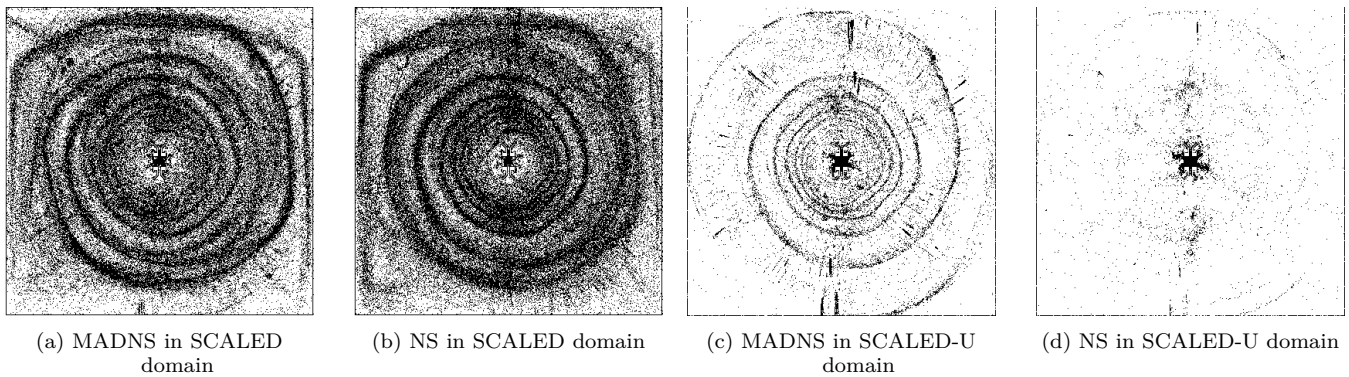
Figure 7: Example exploration of domain over a single evolutionary run for MADNS and NS in bounded and unbounded domains (identical random seeds, $w = 5000$)

annual conference on Genetic and evolutionary computation, pages 103–110. ACM, 2010.

[12] J. Lehman and K. O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011.

[13] J. Lehman and K. O. Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 211–218. ACM, 2011.

[14] A. Liapis, G. N. Yannakakis, and J. Togelius. Enhancements to constrained novelty search: Two-population novelty search for generating game content. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 343–350. ACM, 2013.

[15] J.-B. Mouret. Novelty-based multiobjectivization. In *New horizons in evolutionary robotics*, pages 139–154. Springer, 2011.

[16] J.-B. Mouret and S. Doncieux. Encouraging behavioral diversity in evolutionary robotics: An empirical study. *Evolutionary computation*, 20(1):91–133, 2012.

[17] M. Preuss, A. Liapis, and J. Togelius. Searching for good and diverse game levels. In *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*, pages 1–8. IEEE, 2014.

[18] S. Risi, S. D. Vanderbleek, C. E. Hughes, and K. O. Stanley. How novelty search escapes the deceptive trap of learning to learn. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 153–160. ACM, 2009.

[19] D. Smith, L. Tokarchuk, and C. Fernando. Evolving diverse strategies through combined phenotypic novelty and objective function search. In *Applications of Evolutionary Computation*, pages 344–354. Springer, 2015.

[20] D. Smith, L. Tokarchuk, and G. Wiggins. Exploring conflicting objectives with madns: Multiple assessment directed novelty search. In *Proceedings of the Companion Publication of the 2016 Annual Conference on Genetic and Evolutionary Computation*, GECCO Companion '16. ACM, 2016.

[21] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.