Is a Self-Adaptive Pareto Approach Beneficial for Controlling Embodied Virtual Robots?

Jason Teo and Hussein A. Abbass

Artificial Life and Adaptive Robotics (A.L.A.R.) Lab
School of Computer Science, University of New South Wales
Australian Defence Force Academy Campus, Canberra, Australia
{j.teo,h.abbass}@adfa.edu.au

Abstract. A self-adaptive Pareto Evolutionary Multi-objective Optimization (EMO) algorithm is proposed for evolving controllers for a virtually embodied robot. The main contribution of the self-adaptive Pareto approach is its ability to produce controllers with different locomotion capabilities in a single run, therefore reducing the evolutionary computational cost significantly. The aim of this paper is to verify this hypothesis.

1 Methods

Creature Morphology: The creature is a basic quadruped with 4 short legs, with dimensions 4 x 1 x 2cm for the torso and 1 x 1 x 1cm for a limb. A leg has two limbs connected through a hinge joint and the upper limb is connected to the torso via a similar joint. Hinges rotate between 0 to 1.57 radians. Each hinge is actuated with a motor controlled using an artificial neural network (ANN).

Genotype Representation: The genotype encodes both the ANN weights and active hidden units using a real-valued matrix and binary vector respectively.

Experiment 1: SPANN – A Self-Adaptive Pareto EMO Algorithm: A modified version of the Self-adaptive Pareto Differential Evolution algorithm (SPDE) [1] is used. SPDE uses elitism through breeding children only from the current Pareto set. It uses a *differential evolution* (DE) crossover operator with the fixed step in the original DE algorithm replaced with a Gaussian step in the SPDE algorithm. The evolutionary parameters are as follows: 1000 generations, 30 individuals, maximum of 15 hidden units, 500 timesteps and 10 repeated runs.

Experiment 2: A Hand-Tuned EMO Algorithm: In this set of experiments, we used an EMO algorithm with user-defined crossover and mutation rates rather than self-adapting parameters in SPANN. Apart from the non-self-adapting crossover and mutation rates, the hand-tuned EMO algorithm is otherwise similar to SPANN in all other respects. Three different crossover and mutation rates were used: 10%, 50% and 90% in both cases giving a total of 9 different combinations. All other evolutionary and simulation parameters remain the same.

Experiment 3: A Weighted Sum EMO Algorithm: Here, we used an EMO algorithm with a single-objective that combined the two objectives using a weighted sum. Apart from the change to the manner in which the objectives

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2724, pp. 1612–1613, 2003.

are evaluated, the weighted sum EMO algorithm is otherwise similar to SPANN in all other respects. 10 different values were used for the relative weights. A $(\lambda + \mu)$ strategy is used where the 15 best individuals of the population are carried over to the next generation intact. All other parameters remain the same.

Experiment 4: A Single-Objective Evolutionary Algorithm (EA): Finally, we used a conventional EA which optimizes only one objective of maximizing the locomotion distance achieved by the ANN controller while keeping the hidden layer size fixed. As in the weighted sum EMO algorithm, the $(\lambda + \mu)$ strategy is used in this single-objective EA. Sixteen separate sets of evolutionary runs were conducted corresponding to each one of the different number of hidden units ranging from 0 to 15, which is the range allowed in the multi-objective runs.

2 Results and Discussions

A summary of the results are presented in Table 1. The total computational cost is estimated using the total number of hidden unit activations registered during the search process for each algorithm since most of the computational time is spent on the evaluation of evolved genotypes representing different ANN controllers within the physics-based simulator. The total computational cost (C) will differ between different algorithms as a function of the number of hidden unit activations required to evaluate the fitness of each newly generated genotype (A), the number of new genotypes generated per evolutionary run (G) and the number of evolutionary runs per algorithm (R), as given by $C = A \times G \times R$.

Table 1. Comparison of best locomotion distance obtained and corresponding total computational cost using SPANN against all other algorithms

Algorithm	Locomotion Distance	$+/-$ % of SPANN	Computational Cost	$+/-$ % of SPANN
SPANN	17.6994	-	909,520,500	-
Hand-Tuned EMO	19.5051	+15.9%	7,529,814,400	+727.9%
Weighted Sum EMO	21.8228	+23.3%	3,073,867,500	+238.0%
Single-Objective EA	22.4069	+26.5%	1,441,441,000,000	+15748.4%

In summary, although controllers evolved using the hand-tuned, weighted sum and single-objective algorithms achieved higher locomotion distances, the trade-off in terms of computational cost was staggeringly high compared to SPANN. Hence, the self-adaptive Pareto approach adopted in SPANN is computationally efficient while producing sufficiently good locomotion controllers.

References

1. Hussein A. Abbass. The self-adaptive Pareto differential evolution algorithm. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC2002)*, volume 1, pages 831–836, Piscataway, NJ, 2002. IEEE Press.