Chapter 6

Verifying the Self-Adaptive Pareto EMO

In this chapter, we employ more conventional methods of evolutionary optimization for the generation of artificial creature controllers and then compare these results against the self-adaptive Pareto approach of the SPANN EMO algorithm. Three evolutionary optimization algorithms are used here, namely a hand-tuned EMO algorithm, a weighted sum EMO algorithm and a single-objective evolutionary optimization algorithm. The objectives of these comparisons are firstly to elucidate the effectiveness of using these conventional algorithms for generating high quality locomotion controllers and secondly whether the advantages of the self-adaptive Pareto approach are truly beneficial against these more common methods of evolutionary optimization. A test of redundancy present in the ANN controllers evolved using SPANN is also conducted and compared against the ANN controllers evolved using the conventional EAs mentioned above. Finally, a comparison of SPANN is made against NSGA-II, a well-known and state-of-the-art Pareto EMO algorithm.

6.1 A Hand-Tuned EMO Algorithm

6.1.1 Experimental Setup

In this set of experiments, we used an EMO algorithm with user-defined crossover and mutation rates rather than self-adapting parameters in the SPANN algorithm (similar to the MPANN algorithm (Abbass 2001) but without back-propagation). Apart from the non-self-adapting crossover and mutation rates, the hand-tuned EMO algorithm is otherwise similar to the SPANN algorithm in all other respects. The NNType3 architecture was used since it provided the best overall results among the different controller architectures (see Section 5.6). Three different crossover rates **(c)** and mutation rates **(m)** were used: 10%, 50% and 90% for both rates giving a total of 9 different combinations. As with SPANN, the fitness of each genotype in these experiments was evaluated according to both the f_1 and f_2 objective functions, which measures the locomotion distance achieved and number of hidden units used by the controller respectively as defined in Section 3.4.1. All other evolutionary and simulation parameters remain the same: 1000 generations, 30 individuals, maximum of 15 hidden units, 500 timesteps and 10 repeated runs. In characterizing the fitness landscapes, the individual genotypes were grouped into 6250 discrete intervals over the increased locomotion distance dimension of 25 used in this chapter, in order to maintain the same frequency distribution's interval length as those previously used in Chapters 4 and 5.

6.1.2 Results and Discussion

In this section, we discuss the solutions produced using the hand-tuned EMO algorithm in terms of the different crossover and mutation rates used during evolution. The best Pareto solutions for locomotion distance obtained from conducting the evolutionary optimization process using user defined rates for the genetic operators are presented in Table 6.1. The highest overall best locomotion distance of 19.5 was achieved using a crossover rate of 10% and a mutation rate of 50% while the lowest overall best locomotion distance of 14.9 was obtained using

Crossover Rate	Mutation Rate	Overall Best Locomotion Distance	Average Best Locomotion Distance ± Standard Deviation	No. of Hidden Units
10%	10%	17.1071	13.5192 ± 2.7845	3.2 ± 1.8
10%	50%	19.5051	14.1158 ± 2.6535	6.6 ± 2.1
10%	90%	14.9493	13.2843 ± 1.8225	8.1 ± 2.7
50%	10%	16.2272	14.1268 ± 2.1286	3.0 ± 2.4
50%	50%	18.5638	15.3819 ± 2.3195	6.5 ± 1.7
50%	90%	16.3347	13.1881 ± 1.4715	7.7 ± 2.5
90%	10%	18.5980	14.1511 ± 2.4721	3.4 ± 1.8
90%	50%	15.8766	13.1978 ± 1.6447	6.1 ± 1.4
90%	90%	15.9395	12.1653 ± 2.3799	6.8 ± 2.9

Table 6.1: Comparison of best locomotion distance for Pareto solutions found over 10 independent runs using the hand-tuned EMO algorithm with different crossover and mutation rates.

a crossover rate of 10% and a mutation rate of 90%. The best result in terms of average best locomotion distance achieved was obtained using a crossover rate of 50% and mutation rate of 50% while the worst overall result was obtained using a crossover rate of 90% and mutation rate of 90%. This suggests that a low to medium crossover coupled with a medium mutation rate provided better results when self-adaptation was not used in the EMO algorithm and conversely, a high mutation rate seemed to provide lower quality results for locomotion distance. In terms of optimizing the hidden layer, the crossover rate did not seem to affect the results while a low mutation rate of 10% consistently gave the smallest hidden layer for the evolved controllers of around 3.2 hidden units compared to higher mutation rates of 50% and 90%.

The evolution of the Pareto solution for best locomotion distance using the hand-tuned EMO algorithm for 10 runs over 1000 generations is shown in Figure 6.1 for four of the nine different combinations of crossover and mutation rates. In general, the characteristics of these convergence graphs were similar to that obtained using SPANN (Figure 5.3.4) in that the progression of fitter solutions being discovered was relatively smooth over time. The top two graphs (Figures 6.1.1 & 6.1.2) are representative of these runs. However, there were also combinations of crossover and

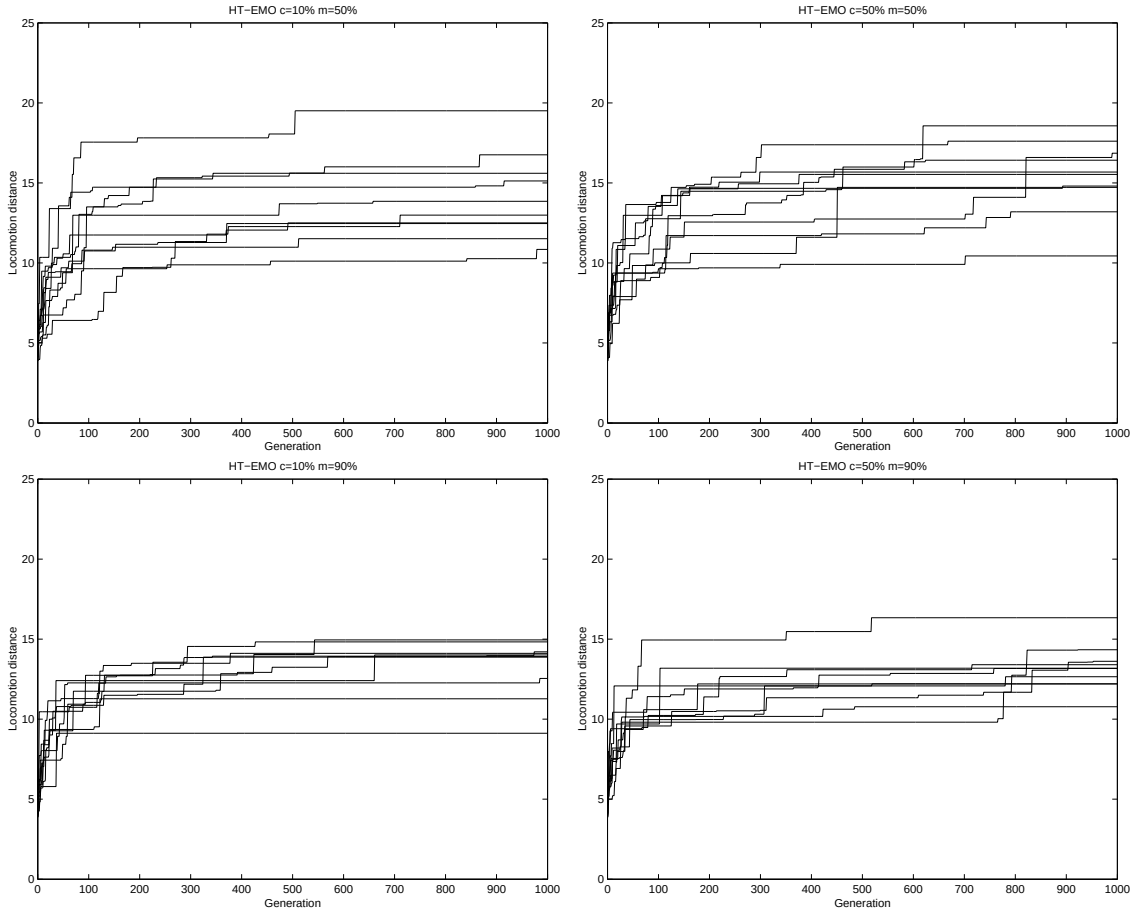

Figure 6.1: Best locomotion distance of Pareto solutions obtained over 1000 generations for 10 runs using the hand-tuned EMO algorithm with 1. c=10% m=50% (top left), 2. c=50% m=50% (top right), 3. c=10% m=90% (bottom left), 4. c=50% m=90% (bottom right). X-axis: Generation, Y-axis: Locomotion distance. Additional graphs can be found in the accompanying CD-ROM.

mutation rates where the improvement of the solutions was much less smooth causing large plateau regions. This phenomenon can be seen in the bottom two graphs (Figures 6.1.3 & 6.1.4), which are highly reminiscent of the graph obtained using hill-climbing (Figure 4.8.4), suggesting that the algorithm may have become stuck in a local optimum for some of the runs. This is likely to be due to the limitation of not being able to change the mutation and crossover rates during the evolutionary optimization process, which may be beneficial in escaping from deep local optima.

As such, these results suggest that self-adaptation, such as that present in SPANN, may be a desirable feature not merely for reducing the computational runs required to test out hand-tuned crossover and mutation rates but also for avoiding premature convergence to less optimal solutions.

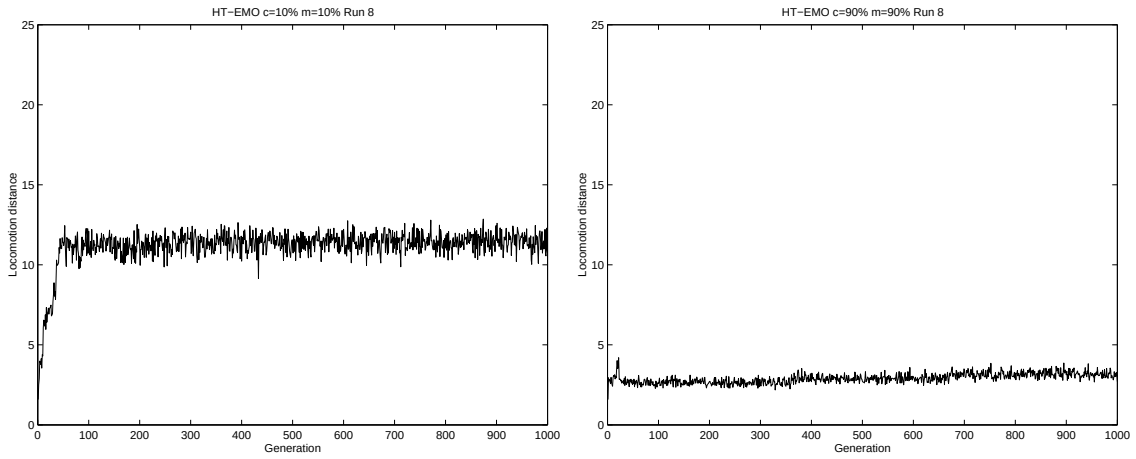

Figure 6.2: Mean locomotion distance of population over 1000 generations using the eighth seed for the hand-tuned EMO algorithm with 1. c=10% m=10% (left), 2. c=90% m=90% (right). X-axis: Generation, Y-Axis: Locomotion distance. Additional graphs can be found in the accompanying CD-ROM.

Figure 6.2 depicts the mean locomotion distance and Figure 6.3 depicts the standard deviation for locomotion distance of the population as it evolved over 1000 generations. The graphs depicted are representative of the trends observed in a large majority of the runs. The average population fitness in terms of locomotion distance remained constant within a fixed range after the initial large jump early during evolution as shown in Figures 6.2.1 and 6.2.2. The two most common trends in the standard deviation of the population were remaining fairly constant within a certain range as shown in Figure 6.3.1 and increasing slightly over time as shown in Figure 6.3.2. Lower crossover and mutation rates seemed to produce a higher population average (Figure 6.2.1) compared to higher crossover and mutation rates (Figure 6.2.2). On the other hand, higher crossover and mutation rates appeared to produce less changes to the standard deviation of the population over time (Figure

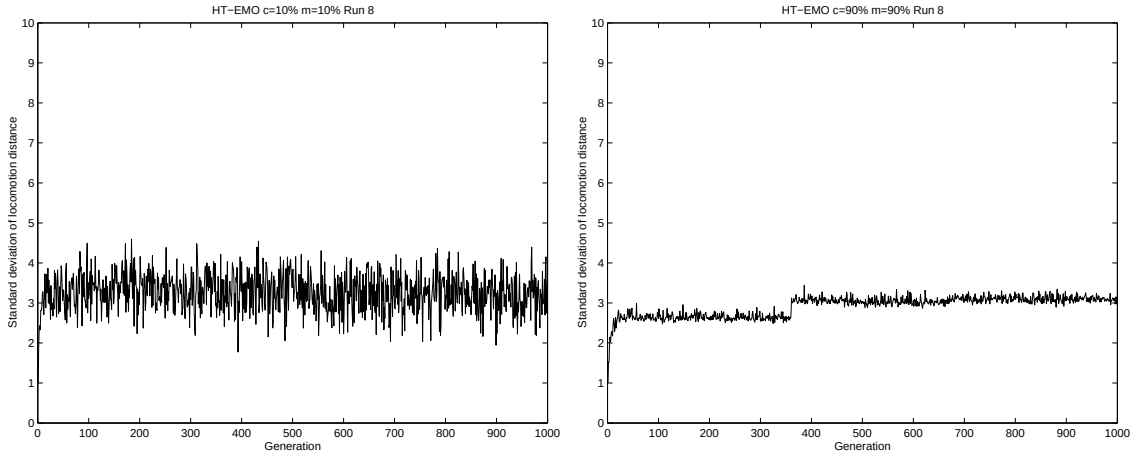

Figure 6.3: Standard deviation for locomotion distance of population over 1000 generations using the eighth seed for the hand-tuned EMO algorithm with 1. c=10% m=10% (left), 2. c=90% m=90% (right). X-axis: Generation, Y-Axis: Standard deviation of locomotion distance. Additional graphs can be found in the accompanying CD-ROM.

6.3.2) compared to lower crossover and mutation rates (Figure 6.3.1). These last two observations suggest that higher crossover and mutation rates were less efficient at finding fitter solutions because the large changes being applied to the genotype at every generation did not allow evolution a chance to discover and maintain a good set of basic genes. Consequently, the solutions being sampled using high crossover and mutation rates were similarly low in fitness resulting in the lower population means and smaller changes in the standard deviations over time.

6.1.3 Search Space Characterization

The distribution of genotypes generated using the hand-tuned EMO algorithm is plotted in Figure 6.4 in terms of locomotion distance and number of hidden units used in the ANN. Although the distribution of solutions across both objective spaces were more uniform compared to that obtained using random search (Figure 4.1.4), hill-climbing (Figure 4.5.4) and random walk (Figure 4.9.4), they were much less uniform and more clustered compared to the distribution obtained using

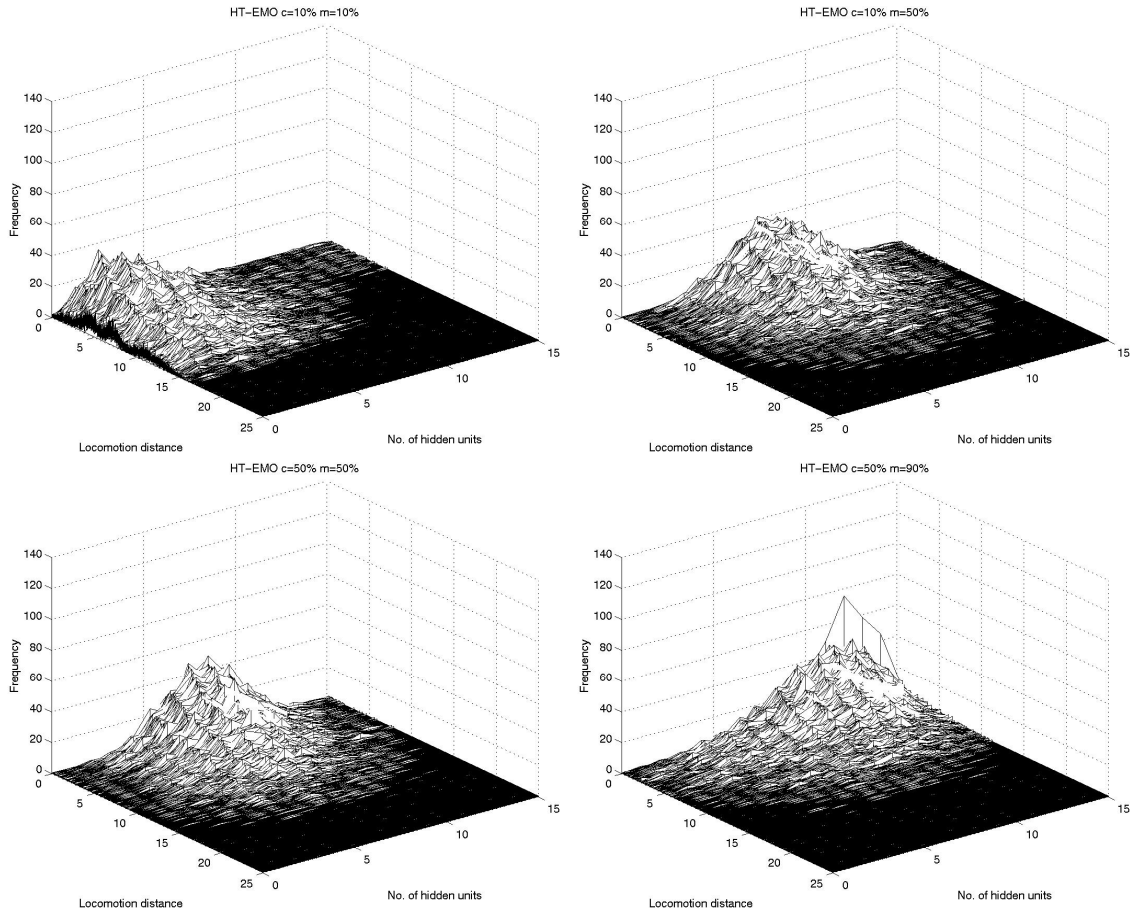

Figure 6.4: Frequency distribution of solutions obtained using the hand-tuned EMO algorithm with 1. c=10% m=10% (top left), 2. c=10% m=50% (top right), 3. c=50% m=50% (bottom left), 4. c=50% m=90% (bottom right). X-axis: Locomotion distance, Y-axis: No. of hidden units, Z-axis: Frequency. Additional graphs can be found in the accompanying CD-ROM.

SPANN (Figure 5.6.4). From the general features observed in these graphs, the distributions most similar to SPANN were those generated using a crossover rate of 10% and mutation rate of 50% (Figure 6.4.2) as well as a crossover rate of 50% and mutation rate of 50% (Figure 6.4.3). The main difference between the hand-tuned EMO algorithm and SPANN is that the solutions in the hand-tuned EMO were mainly clustered around the lower fitness regions of the search space, a large majority of which yielded only between 0 and 10 units of locomotion distance compared

to SPANN where a more even distribution could be seen to extend to fitness regions of 15 units of locomotion distance.

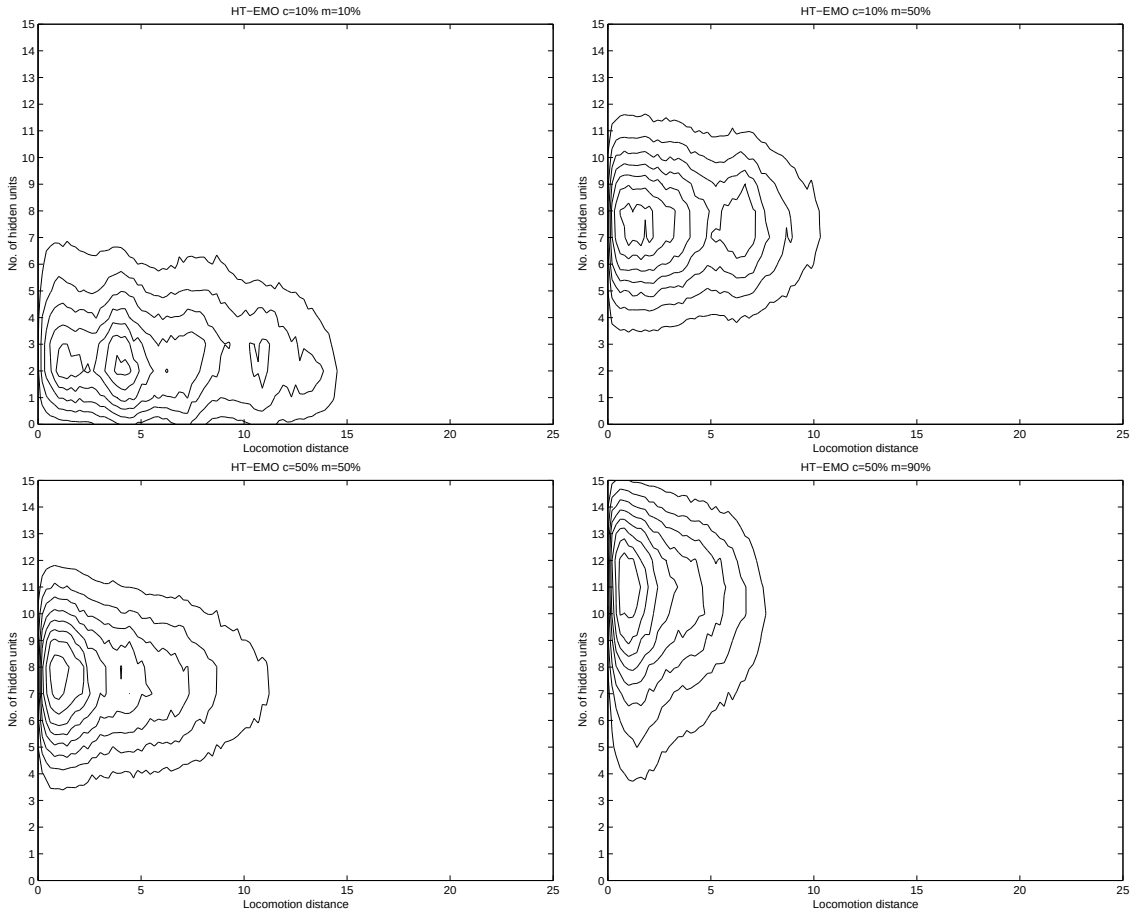

Figure 6.5: Contour graphs of frequency distribution of solutions obtained using the hand-tuned EMO algorithm with 1. c=10% m=10% (top left), 2. c=10% m=50% (top right), 3. c=50% m=50% (bottom left), 4. c=50% m=90% (bottom right). X-axis: Locomotion distance, Y-axis: No. of hidden units. Additional graphs can be found in the accompanying CD-ROM.

The contour graphs in Figure 6.5 illustrate the distribution of solutions across the two objectives of minimizing the hidden layer and maximizing locomotion distance. A number of interesting features emerged in these contour graphs. Firstly, the mutation rate significantly affected the range of genotypes generated in terms of the number of hidden units used in the controller. The controllers evolved using

the lowest mutation rate of 10% centered around a usage of between 2 and 3 hidden units (Figure 6.5.1). When the mutation rate was increased to 50%, the solutions now centered around a higher usage of between 7 and 8 hidden units (Figures 6.5.2 & 6.5.3), and furthermore, in the highest setting of the mutation rate at 90%, the solutions clustered around controllers that used between 10 and 12 hidden units (Figure 6.5.4). The contour features most similar to SPANN (Figure 5.8.4) again could be seen when the crossover rate was set at 10% and mutation rate at 50% (Figure 6.5.2) as well as at a crossover rate of 50% and mutation rate of 50% (Figure 6.5.3), although in both these cases the spread of solutions over the objective spaces were less uniformly distributed. Interestingly, the movement of solutions to larger hidden layer sizes produced lower locomotion capabilities. The underlying fitness landscape may have become more rugged as the size of hidden layer increased and as previously postulated, the non-self-adapting crossover and mutation rates may have represented a severe limitation in allowing the algorithm to move through these landscapes, subsequently causing the optimization process to become trapped around sub-optimal regions of the search space.

The probability density function of solutions obtained using the hand-tuned EMO algorithm is illustrated in Figure 6.6. The probability density curves show that a low crossover rate of 10% and low mutation rate of 10% provided the best distribution of solutions across the locomotion objective space (Figure 6.6.1). The cumulative curve shows that the probability of encountering fitter solutions decreased noticeably as the mutation rate was increased to 50% (Figures 6.6.2 & 6.6.3) and especially 90% (Figure 6.6.4), where the probability of encountering a solution decreased to 0 beyond a fitness of only around 11.

The search space characterization of the hand-tuned EMO algorithm showed significantly different characteristics compared to SPANN, especially when very high and very low combinations of crossover and mutation rates were used. As previously discussed, the correct choice of these two rates by the user is paramount in obtaining reasonably good results when using this form of the EMO algorithm. This analysis also showed that the use of self-adaptive crossover and mutation rates

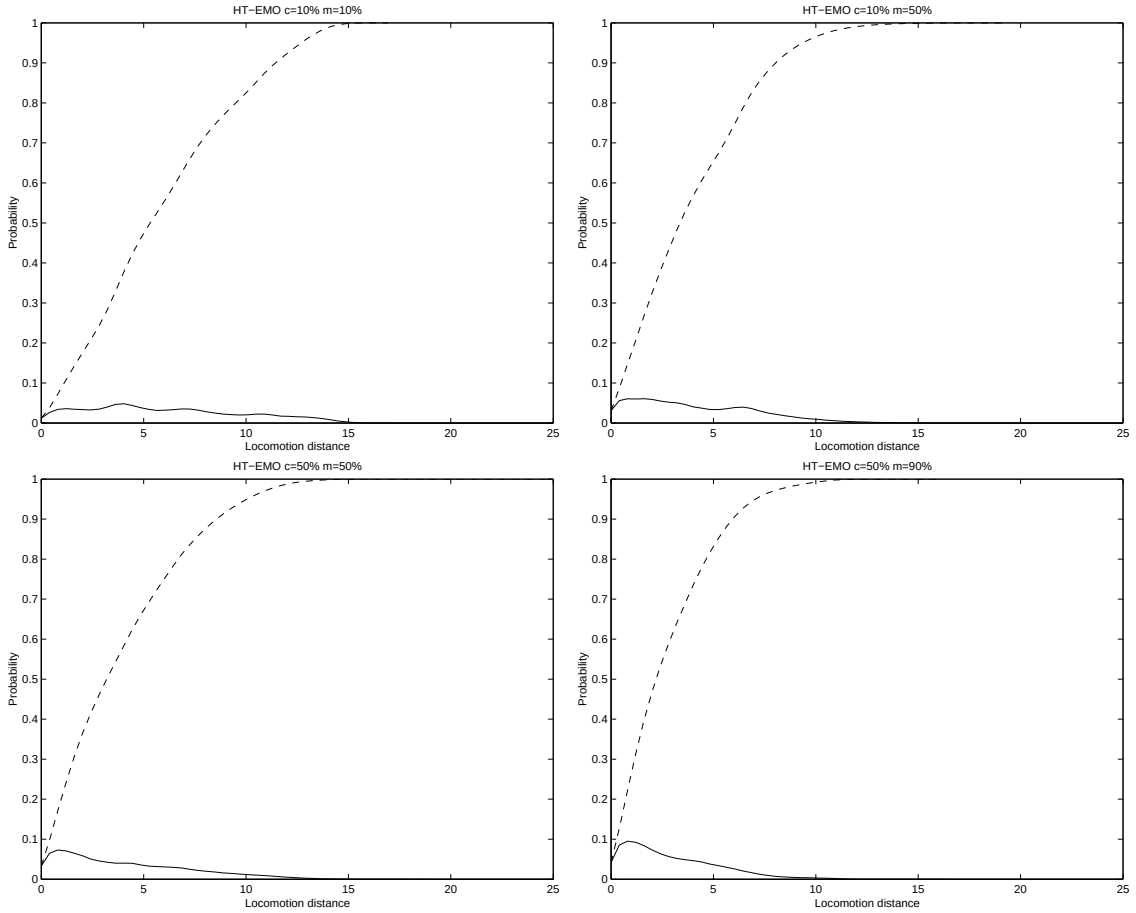

Figure 6.6: Density (solid) and cumulative (dashed) probability distribution of solutions obtained using the hand-tuned EMO algorithm with 1. c=10% m=10% (top left), 2. c=10% m=50% (top right), 3. c=50% m=50% (bottom left), 4. c=50% m=90% (bottom right). X-axis: Locomotion distance, Y-axis: Probability. Additional graphs can be found in the accompanying CD-ROM.

in SPANN allowed the evolutionary search to sample a much larger area of the objective space compared to the non-self-adaptive EMO algorithm and as such was able to perform more effectively in finding good locomotion controllers.

6.2 A Weighted Sum EMO Algorithm

6.2.1 Experimental Setup

For this second set of experiments, we used a single objective that combined the two objectives f_1 and f_2 using a weighted sum rather than a true Pareto approach as found in the SPANN algorithm, which distinctly separates the two objectives when assigning fitness values to individuals in the population. As in the comparison between the hand-tuned versus self-adaptive EMO algorithms in the previous section, the NNType3 architecture was used in this set of experiments for the same reason that it provided the best overall result among the different architectures as reported in Section 5.6. The weighting of the individual objectives was done in a relative manner using a parameter denoted by γ. In order to combine the two objectives f_1, which is the maximization of the locomotion distance, and f_2, which is the minimization of the number of hidden units used in the ANN, into a single weighted sum fitness function, these objectives needed to be unified in terms of their direction of optimization. Firstly, the locomotion distance objective f_1 was re-defined to be a minimization problem

$$f_1' = 100.0 - f_1 \tag{6.1}$$

which yielded the minimization of the overall weighted sum function as follows:

$$\gamma \times f_1' + (1 - \gamma) \times f_2 \tag{6.2}$$

However, we chose to convert the overall weighted sum optimization problem into one of maximization to maintain consistency when presenting the solutions in terms of locomotion distance achieved by the best evolved controllers. Hence the final weighted sum objective function is given by

$$f(overall) = 100.0 - [(\gamma \times f_1') + ((1 - \gamma) \times f_2)] \tag{6.3}$$

where $f(overall)$ represents the weighted fitness and γ is the relative weight parameter. The unification of the two objectives f_1 and f_2 could have been similarly

achieved by converting f_2 into a maximization problem. 10 different values were used for γ ranging from 10% to 100% in increments of 10%. No setup for $\gamma = 0\%$ was used since in this case, no optimization would be performed on f_1, which means that in such an evolutionary run, there will be no pressure for the controller to develop any locomotion ability whatsoever. On the other hand, $\gamma = 100\%$ would result in an evolutionary run with no pressure towards minimizing the number of hidden units since no optimization would be performed on f_2. All this means is that the hidden layer in the controller is completely free to use any number of hidden units for optimizing the locomotion behavior, therefore this setup is retained in the experiments.

An approach similar to the $(\mu + \lambda)$ evolutionary strategy is used where the 15 best individuals of the population are carried over to the next generation without any modification to the genotype at all. This is to allow a setup similar to SPANN, where the upper bound on the number of Pareto solutions is simply $1 + 15$, the maximum number of hidden units allowed. The crossover and mutation operators function as in SPANN and the rates for these genetic operators are also self-adaptive. The only real difference between SPANN and the weighted sum method is the objective function and therefore, the selection mechanism. For all other parameters, they remain the same as in all other experiments: 1000 generations, 30 individuals, maximum of 15 hidden units, 500 timesteps and 10 repeated runs. As in the hand-tuned EMO algorithm search space characterization, the individual genotypes generated were grouped into 6250 discrete intervals to cater for the increased locomotion distance dimension.

6.2.2 Results and Discussion

The results obtained from using the weighted sum method for conducting the EMO process are given in Table 6.2. In the analysis of this set of results, we first present the solution fitness in terms of the actual weighted sum value that is used for the selection process. Then, we decompose the weighted sum fitness into the two separate objectives of locomotion distance and number of hidden units, and

γ	Weighted Sum Value ± Standard Deviation	Overall Best Locomotion Distance	Locomotion Distance ± Standard Deviation	No. of Hidden Units
10%	90.9857 ± 0.1428	12.3513	9.8571 ± 1.4277	0.0 ± 0.0
20%	82.0122 ± 0.4539	14.5964	10.4613 ± 2.6883	0.1 ± 0.3
30%	72.4592 ± 0.3774	9.8821	8.4306 ± 1.3288	0.1 ± 0.3
40%	63.4604 ± 0.6815	12.3985	9.4011 ± 2.1017	0.5 ± 0.8
50%	55.2962 ± 1.4375	15.8411	11.3924 ± 3.0330	0.8 ± 0.9
60%	46.6677 ± 1.9308	16.4046	12.1794 ± 2.9865	1.6 ± 1.0
70%	38.6314 ± 1.2593	17.9004	13.7448 ± 2.4376	3.3 ± 1.9
80%	30.4217 ± 1.6079	17.7011	14.0521 ± 2.3034	4.1 ± 1.5
90%	23.0408 ± 1.8146	18.1530	15.1119 ± 1.9977	5.6 ± 1.9
100%	15.2829 ± 3.6578	21.8228	15.2829 ± 3.6578	8.1 ± 1.5

Table 6.2: Best solutions obtained over 10 independent runs using the weighted sum EMO algorithm with different weights for the two objectives. γ = relative weight parameter.

present the analysis from these distinct points of view. Firstly, it is apparent that the correct combination of weights played a critical role in obtaining good results from the evolutionary optimization runs when both locomotion distance and hidden layer size are to be considered simultaneously in a single weighted objective. Setting γ to between 60% and 90% seemed to provide a good trade-off between achieving a reasonably good locomotion capability and relatively small hidden layer size. Setting $\gamma = 100\%$, which places all the optimization pressure on the locomotion component, yielded the highest overall and average best locomotion distance. However, this correspondingly resulted in the highest average of hidden units used in the evolved ANNs since there was no pressure to minimize the hidden layer at all. On the other hand, the lowest average of hidden units used was obtained when γ was set to 10%, although the average best locomotion distance achieved was much lower at only around 9.9 by virtue of the very large weighting assigned to the hidden unit component and correspondingly small weighting assigned to the locomotion component. Although it was expected that the average best locomotion distance obtained would increase monotonically as γ increased, this was not the case in the runs where the parameter was set to 30% and 40%. This was likely due to the fact that controllers with very small hidden layer sizes dominated the elite solutions and

subsequently caused the optimization process to become stuck in a local optima centered around these small-sized controllers with limited locomotion capabilities. Further analysis of this phenomenon is given in the following paragraph as well as in Section 6.2.3.

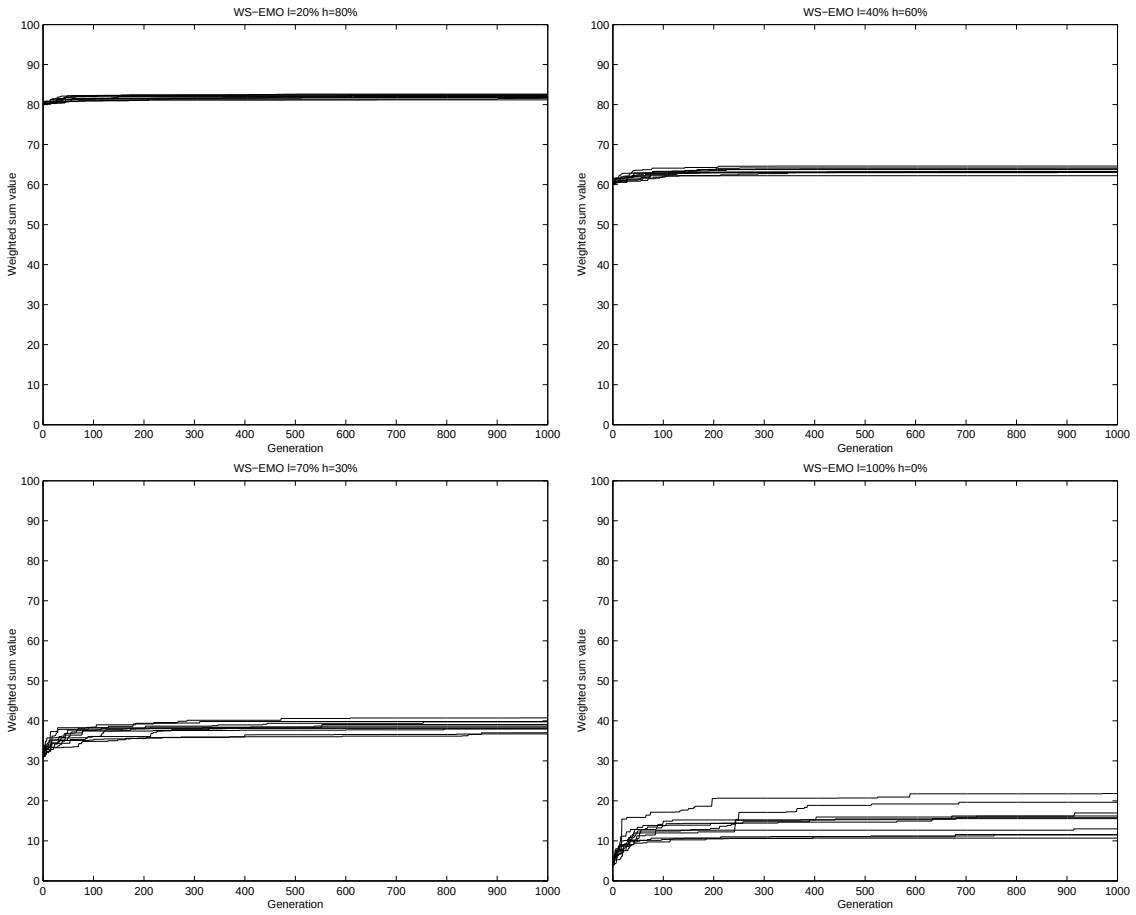

Figure 6.7: Best solutions obtained over 1000 generations for 10 runs using the weighted sum EMO algorithm with 1. $\gamma = 20\%$ (top left), 2. $\gamma = 40\%$ (top right), 3. $\gamma = 70\%$ (bottom left), 4. $\gamma = 100\%$ (bottom right). X-axis: Generation, Y-axis: Weighted sum value. Additional graphs can be found in the accompanying CD-ROM.

The evolution of the best solution using the weighted sum EMO algorithm for 10 runs over 1000 generations is shown in Figure 6.7 for four of the ten different combinations of weights assigned to the respective objectives. These graphs depict

the progression of the best solutions in terms of the actual weighted fitness value as evaluated using the weighted sum objective. The fixed weighting of the objectives appeared to have a significant impact on the improvement of the best solutions over time. When γ was set to low values as depicted in the top two graphs, the solutions were only able to improve over a highly constrained weighted value (Figures 6.7.1 & 6.7.2). This again was likely due to controllers using small numbers of hidden units being assigned high fitness values and hence dominating the elite solutions. As the value of γ was increased, the solutions were able to improve over a larger range of weighted values, although this also increased the variations between the best solutions found, as shown by the bottom two graphs (Figures 6.7.3 & 6.7.4). In the following paragraphs, we discuss the convergence of the solutions from the viewpoint of the separate component objectives.

The convergence of the best solution using the weighted sum EMO algorithm for 10 runs over 1000 generations is shown in Figure 6.8 in terms of the locomotion distance for the same four combinations of weights as in Figure 6.7. In the majority of the runs, most of the improvement achieved in terms of locomotion distance occurred very early during evolution, generally around the 100–120th generation. Also, the effect of rewarding controllers with smaller hidden layers can be seen clearly in Figure 6.8.3 where solutions with lower locomotion fitness but using less hidden units were accepted as the current best solution, causing this particular graph to have periods of apparently lower fitness during the convergence process. This in fact occurred because the weighted sum value of certain controllers that achieved greater locomotion distances were actually lower than controllers with less locomotion capabilities, as a result of using more hidden units in the hidden layer of the ANN compared to the less effective locomotion controllers. The progression of the use of hidden nodes over the evolutionary optimization process can be seen in the graphs that follow.

The convergence of the best solution using the weighted sum EMO algorithm for 10 runs over 1000 generations is shown in Figure 6.9 in terms of the number of hidden units used in the controller for the same four combinations of

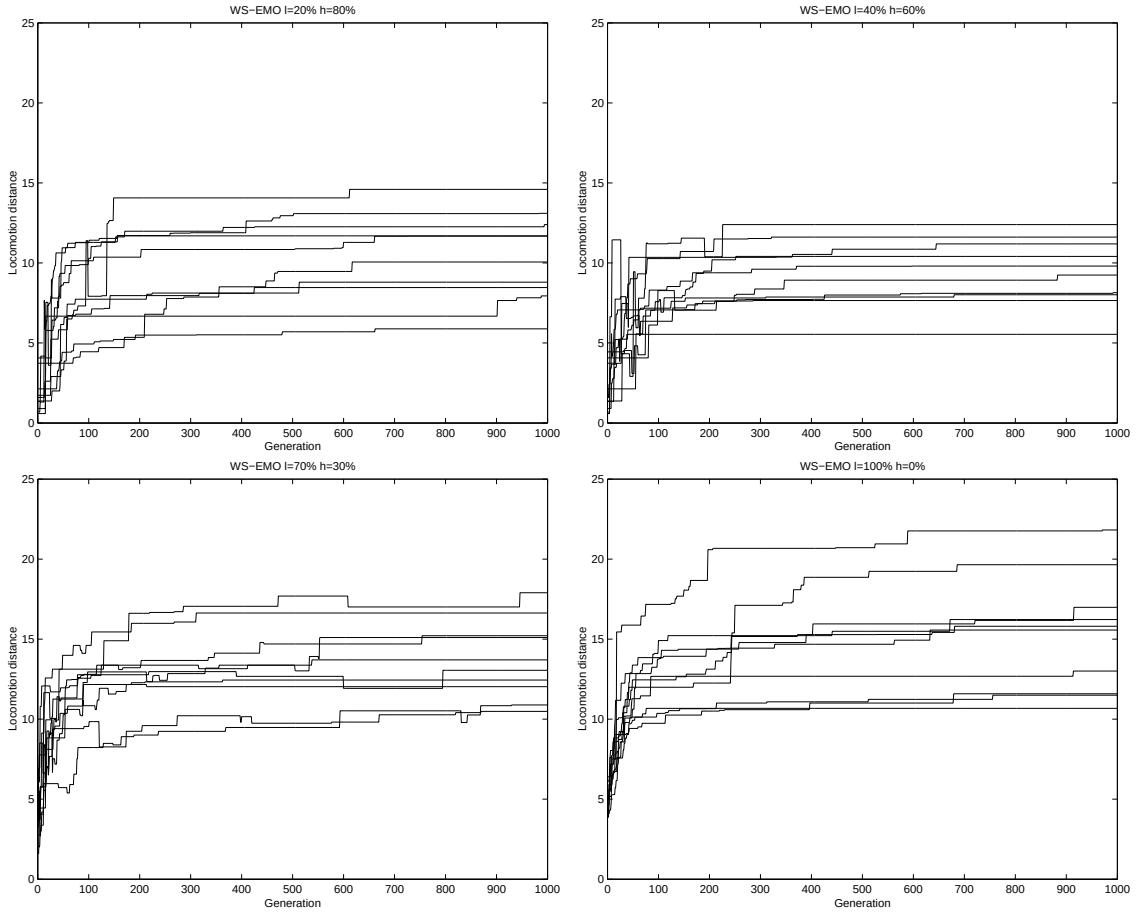

Figure 6.8: Locomotion distance of best solutions obtained over 1000 generations for 10 runs using the weighted sum EMO algorithm with 1. $\gamma = 20\%$ (top left), 2. $\gamma = 40\%$ (top right), 3. $\gamma = 70\%$ (bottom left), 4. $\gamma = 100\%$ (bottom right). X-axis: Generation, Y-axis: Locomotion distance. Additional graphs can be found in the accompanying CD-ROM.

weights as in the previous paragraphs. The effect of assigning the larger proportion of the weighted fitness to minimizing the hidden layer size can be seen clearly in the top two figures (Figures 6.9.1 & 6.9.2). The best solutions in these cases were highly constrained during the majority of the evolutionary process, using only 2 or less hidden nodes in the ANN controller. This consequently limited the ability of the solutions to improve on the locomotion distances achieved as a direct result of being able to only use controllers with very small numbers of hidden nodes. As γ is

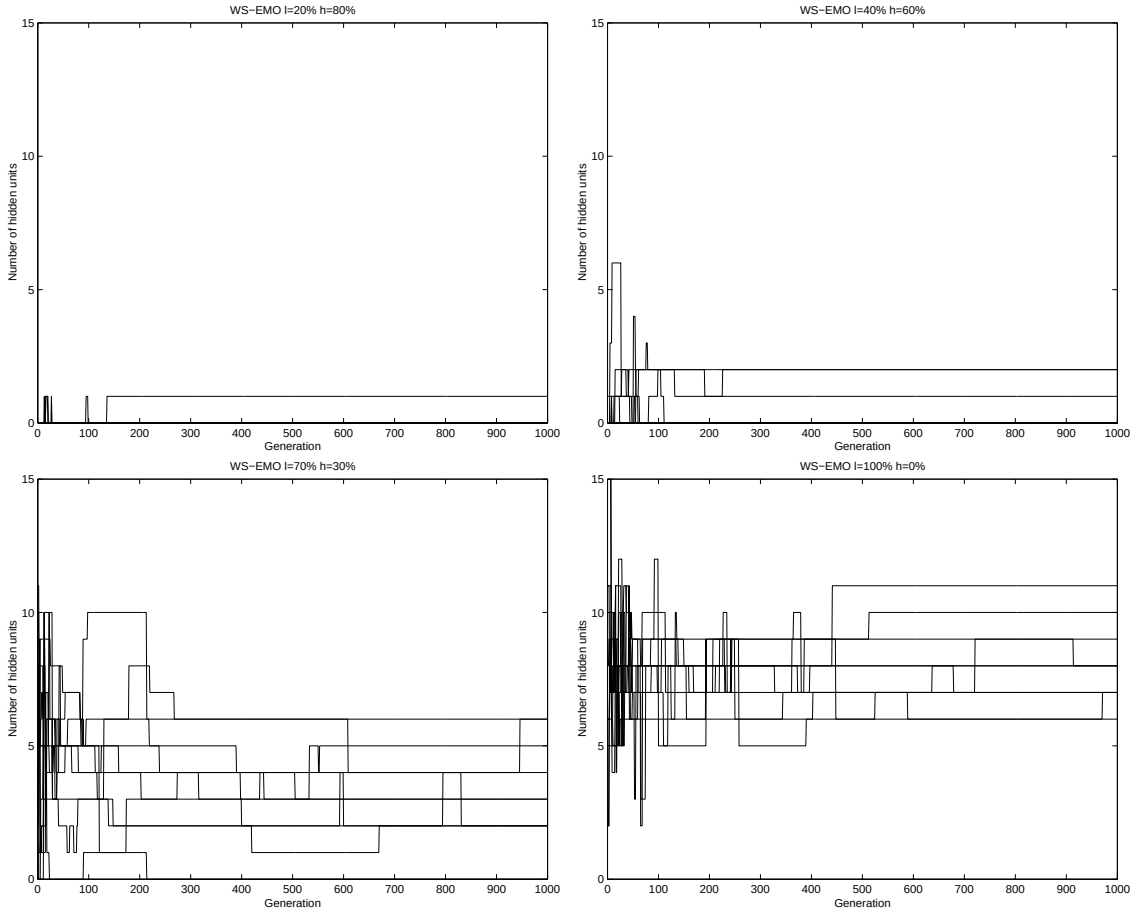

Figure 6.9: Hidden layer size of best solutions obtained over 1000 generations for 10 runs using the weighted sum EMO algorithm with 1. $\gamma = 20\%$ (top left), 2. $\gamma = 40\%$ (top right), 3. $\gamma = 70\%$ (bottom left), 4. $\gamma = 100\%$ (bottom right). X-axis: Generation, Y-axis: Number of hidden units. Additional graphs can be found in the accompanying CD-ROM.

increased to give more weight to the locomotion component, the constraint on the size of the hidden layer is lessened, thereby increasing the algorithm's likelihood of improving on the quality of the locomotion behavior by having more opportunities to experiment with ANN controllers with larger hidden layer sizes.

The mean locomotion distance and standard deviation for locomotion distance of the population as it evolved over 1000 generations using the weighted sum EMO algorithm are illustrated in Figures 6.10 and 6.11 respectively. Higher values

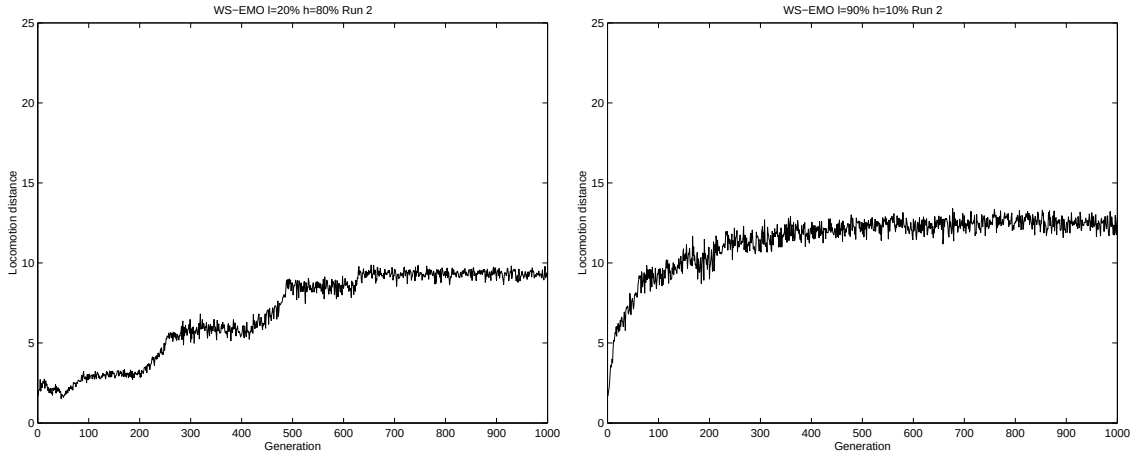

Figure 6.10: Mean locomotion distance of population over 1000 generations using the second seed for the weighted sum EMO algorithm with 1. $\gamma = 20\%$ (left), 2. $\gamma = 90\%$ (right). X-axis: Generation, Y-Axis: Locomotion distance. Additional graphs can be found in the accompanying CD-ROM.

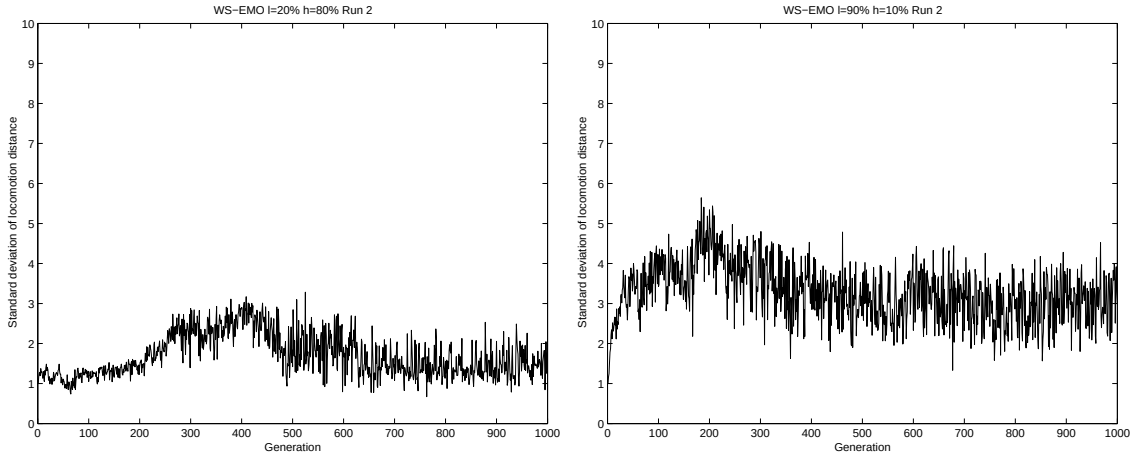

Figure 6.11: Standard deviation for locomotion distance of population over 1000 generations using the second seed for the weighted sum EMO algorithm with 1. $\gamma = 20\%$ (left), 2. $\gamma = 90\%$ (right). X-axis: Generation, Y-Axis: Standard deviation of locomotion distance. Additional graphs can be found in the accompanying CD-ROM.

of γ generally resulted in higher population means, which is expected as more weight is placed on optimizing the locomotion distance over minimizing the hidden layer.

A large majority of the population means reached a fairly constant range after the initial increase in fitness and either remained fixed within a small range or increased slightly in fitness as shown by Figure 6.10.2, which are similar to the trends observed in the hand-tuned EMO algorithm. Another interesting trend that emerged was that of the population mean increasing in a step-wise manner as depicted in Figure 6.10.1 but this only occurred in less than 10% of the runs. This phenomenon was most probably caused by the elite solutions becoming dominated by particular classes of solutions that changed their usage of the number of hidden units in the ANN controller and at the same time achieved distinctly better locomotion capabilities, resulting in short periods of constant fitness followed by significant jumps in fitness. The use of strong elitism in the weighted sum EMO algorithm also produced another common feature in almost all of the runs in that the population mean did not show any significant decrease in fitness over the evolutionary optimization process. This can be explained by the fact that the 15 best individuals representing the elite solutions carried forward from the previous generation will buffer any significant drop in the mean fitness of newly created individuals in the current generation. This observation is supported by the large movements of the standard deviation over time shown in Figure 6.11.2, which is representative of a large majority of the runs. These movements are noticeably larger than the standard deviations observed using the hand-tuned EMO algorithm where only the non-dominated solutions rather than elite solutions were retained. This strong elitism causes two distinct populations to emerge, one in the carried over individuals and another in the newly generated individuals. Consequently, when a number of new individuals are either good solutions similar to the elite solutions or bad solutions far removed from the elite solutions, the standard deviation will correspondingly change very significantly with strong elitism.

6.2.3 Search Space Characterization

The distribution of genotypes generated using the weighted sum EMO algorithm is plotted in Figure 6.12 in terms of locomotion distance and number of

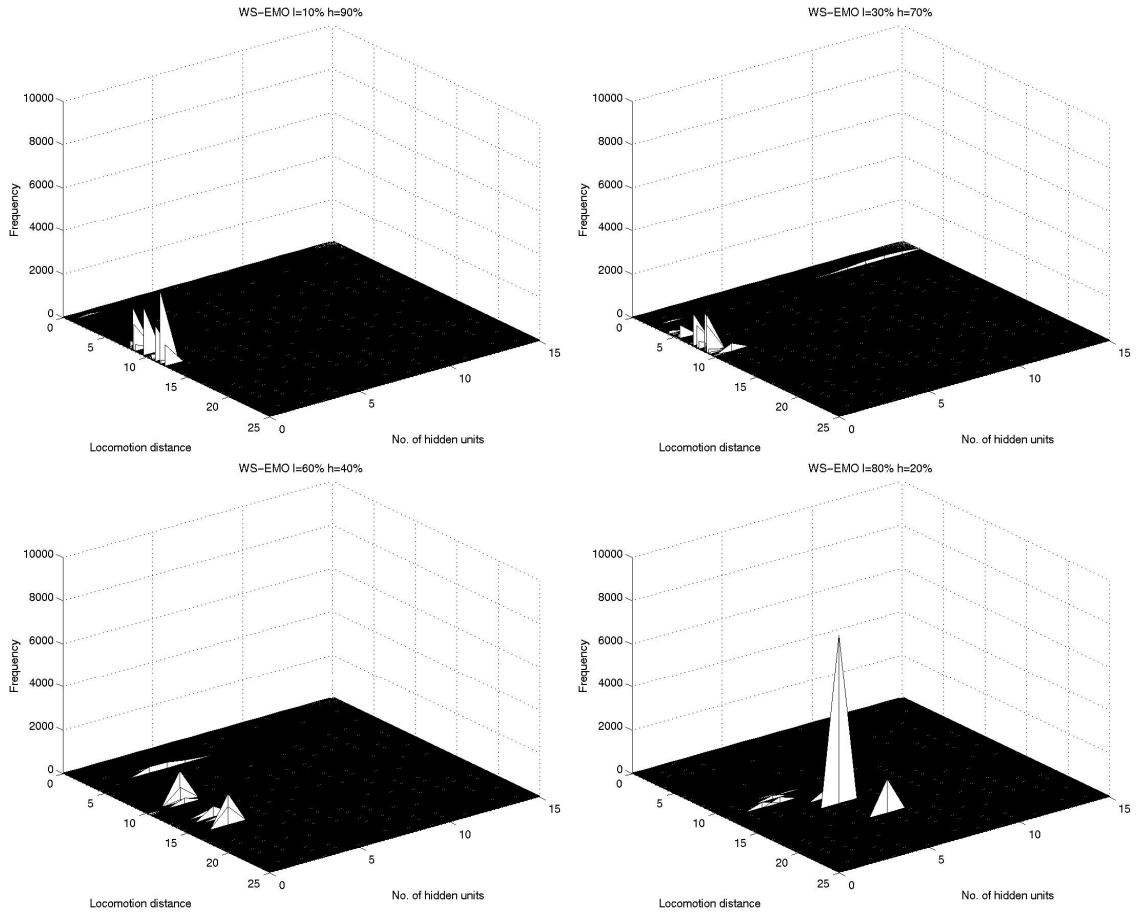

Figure 6.12: Frequency distribution of solutions obtained using the weighted sum EMO algorithm with 1. $\gamma = 10\%$ (top left), 2. $\gamma = 30\%$ (top right), 3. $\gamma = 60\%$ (bottom left), 4. $\gamma = 80\%$ (bottom right). X-axis: Locomotion distance, Y-axis: No. of hidden units, Z-axis: Frequency. Additional graphs can be found in the accompanying CD-ROM.

hidden units used in the ANN. Note that the frequency axis has been expanded from 140 in prior graphs to 10000 to cater for the higher concentrations of genotypes found within a specific range of objective values. Firstly, the distribution of genotypes across the objective spaces were dramatically different compared to SPANN and the hand-tuned EMO algorithm in that the generated genotypes were found to cluster very closely around highly specific values of locomotion distance and number of hidden units. The very significant change to the characteristics of the search

space is likely due to the use of a weighted sum approach coupled with an elitist approach. It is clear from the figures that genetic diversity in terms of the number of hidden units is not equivalent to that achieved with the Pareto approach. Allowing individuals to survive based solely on the weighted sum objective resulted in what can be seen in the figures, where hidden layers with certain numbers of hidden units dominated the evolutionary process. On the other hand, the carrying over of only non-dominated solutions in SPANN and the hand-tuned EMO algorithm leaves more room for variation since each parent is at least entirely different from the other in terms of the size of the hidden layer. Hence, the newly generated individuals can be expected to have greater diversity and consequently sample a larger proportion of the search space.

The contour graphs in Figure 6.13 illustrate the distribution of solutions across the two objectives of minimizing the hidden layer and maximizing locomotion distance. Two trends emerged when using the weighted sum approach in terms of the concentration of solutions across the respective objective spaces. The first trend is that of extremely high concentrations of solutions within very specific areas of the objective space, as evidenced by Figure 6.13.4. This phenomenon occurred when γ was set to 40%, 50%, 70% and 80%. The second group of genotypes had less highly concentrated distributions compared to the first group and had wider sampling of the search space as shown by Figures 6.13.1, 6.13.2 and 6.13.3, although this was still much less compared to the hand-tuned EMO algorithm and especially to SPANN. This supports the earlier observations that the respective weights assigned to the different objectives can significantly affect the behavior of the evolutionary optimization algorithm. As shown by this analysis, some combination of weights will cause the generated solutions to sample only a very limited area of the search space.

The probability density function of solutions obtained using the weighted sum EMO algorithm is illustrated in Figure 6.14. The probability of encountering different classes of solutions in terms locomotion capability varied considerably as different combinations of weights were used to evaluate the generated genotypes. As

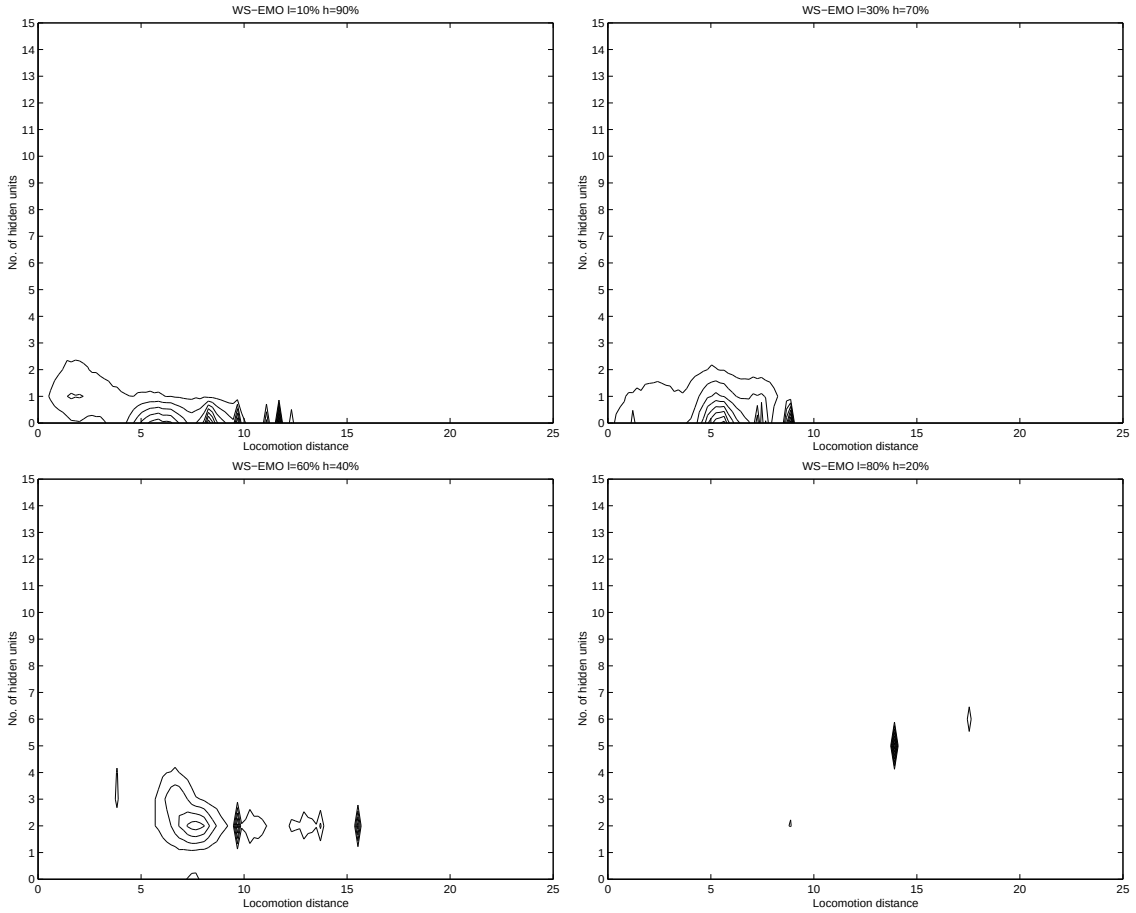

Figure 6.13: Contour graphs of frequency distribution of solutions obtained using the weighted sum EMO algorithm with 1. $\gamma = 10\%$ (top left), 2. $\gamma = 30\%$ (top right), 3. $\gamma = 60\%$ (bottom left), 4. $\gamma = 80\%$ (bottom right). X-axis: Locomotion distance, Y-axis: No. of hidden units. Additional graphs can be found in the accompanying CD-ROM.

expected, as more weight was placed on the locomotion component by increasing γ, the probability density curve could be seen to shift more to the right. However, as pointed out in earlier sections, certain combinations did not perform according to this expectation. Figure 6.14.2 shows that the probability of encountering solutions dropped to 0 as early as 9 units of locomotion distance. This rather poor performance is only slightly better than that achieved using random search (Figure 4.3.4), hill-climbing (Figure 4.7.4) and random walk (Figure 4.11.4) where the probability

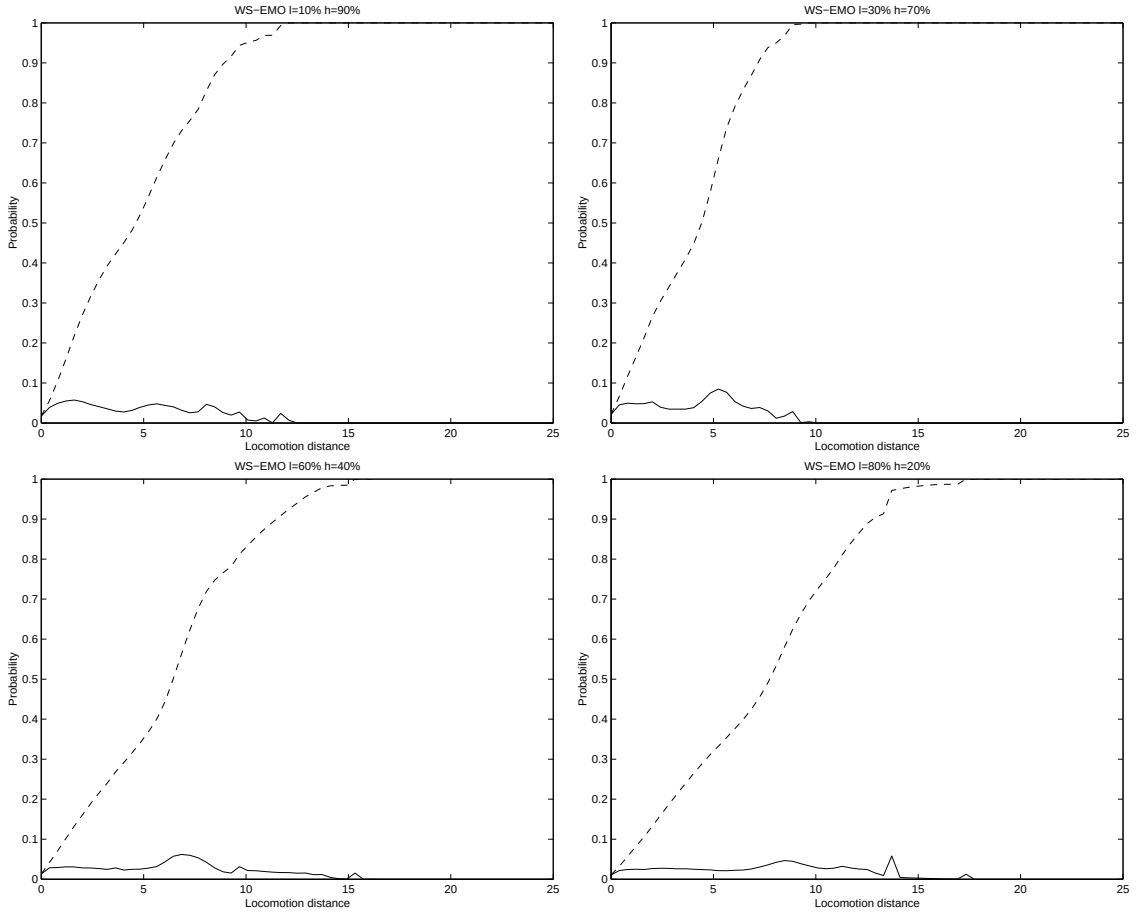

Figure 6.14: Density (solid) and cumulative (dashed) probability distribution of solutions obtained using the weighted sum EMO algorithm with 1. $\gamma = 10\%$ (top left), 2. $\gamma = 30\%$ (top right), 3. $\gamma = 60\%$ (bottom left), 4. $\gamma = 80\%$ (bottom right). X-axis: Locomotion distance, Y-axis: Probability. Additional graphs can be found in the accompanying CD-ROM.

dropped to 0 between 7 to 8 units of locomotion distance.

In summary, the search space characteristics of the EMO algorithm using a weighted sum approach pointed to the fact that although reasonably good sampling of the search space can be achieved with some weight combinations of the respective objectives, dramatically sparse sampling can similarly occur. Moreover, the sampling of the search space for the better combinations of weights were still significantly less uniformly distributed compared to SPANN and even the hand-tuned

EMO algorithm to a lesser extent.

6.3 A Single-Objective EA

6.3.1 Experimental Setup

In the last set of experiments, we used a conventional EA which optimizes only one objective as opposed to optimization of multiple objectives in EMO algorithms such as SPANN. The only objective being optimized in the following evolutionary runs is the locomotion distance achieved by the creature's ANN controller while the size of the hidden layer is kept fixed. Hence, only the f_1 fitness function is used to evaluate the genotypes. Apart from the change of optimizing two objectives to one, the single-objective algorithm is otherwise similar to the SPANN algorithm in all other respects. The crossover and mutation rates are self-adaptive and are identical to their counterparts in SPANN except that crossover and mutation now excludes any changes to the number of hidden units used in the ANN controller since this component is fixed in the single-objective EA. As in previous comparisons, the NNType3 architecture was used in this set of experiments and all other parameters remained the same: 1000 generations, 30 individuals, 500 timesteps and 10 repeated runs. As in the weighted sum EMO algorithm discussed in Section 6.2, the $(\mu + \lambda)$ strategy is used in this single-objective EA where the 15 best individuals of the population are carried over to the next generation without any modification to the genotype at all. Sixteen separate sets of evolutionary runs were conducted corresponding to each one of the different number of nodes used in the hidden layer ranging from 0 to 15, which is the range allowed in the multi-objective runs. As with prior algorithms used in this chapter, the individual genotypes generated were grouped into 6250 discrete intervals for search space characterization.

6.3.2 Results and Discussion

The results obtained from using the single-objective EA for conducting the EMO process are given in Table 6.3. The highest overall best f_1 fitness of 22.4 was

No. of Hidden Units	Overall Best Fitness	Worst of the Best Fitness	Average Best Fitness ± Standard Deviation
0	20.3725	11.6791	15.7516 ± 2.9721
1	19.3005	12.2596	15.1441 ± 2.0260
2	19.8772	9.9934	16.3236 ± 2.7242
3	19.2861	9.5247	15.1532 ± 3.1696
4	20.6868	12.9391	15.2088 ± 2.2106
5	19.9139	11.2756	15.1562 ± 2.8741
6	19.6655	12.7716	16.0317 ± 2.0719
7	17.8093	13.4571	15.8033 ± 1.6159
8	21.6668	12.1226	17.4358 ± 3.2508
9	20.4605	12.1012	15.7375 ± 2.6430
10	19.4172	13.5765	16.1514 ± 2.1318
11	21.6224	9.3723	15.0614 ± 3.5612
12	22.3296	11.6783	15.4287 ± 3.0020
13	17.6432	12.1548	15.0359 ± 1.8909
14	22.4069	10.9295	16.6273 ± 2.8095
15	19.7747	12.5919	15.6150 ± 2.4605

Table 6.3: Best solutions obtained over 10 independent runs using the single-objective EA with different hidden layer sizes.

obtained using a hidden layer of 14 nodes while the lowest overall best fitness of 17.6 was obtained using a hidden layer of 13 nodes. In terms of average best locomotion distance achieved, the setup in which the ANN controller's hidden layer was fixed to use 8 nodes provided the best result while the worst result was obtained when the hidden layer was fixed at 11 nodes. In general, the variations between the best solutions achieved were high when using the single-objective EA. Only 2 out of the 16 different hidden layer setups had standard deviations of less than 2, 10 setups had standard deviations of between 2 and 3 while the remaining 4 setups had standard deviations of more than 3 units distance. The setup which used 11 hidden units also had the highest standard deviation where the difference between the best and worst solutions obtained was 12.2, which is a variation of more than 56% of the overall best fitness achieved using this hidden layer setup. These observations suggest that optimizing the artificial creature's controllers using the single-objective EA is quite unstable and the quality of solutions obtained can vary greatly between different runs.

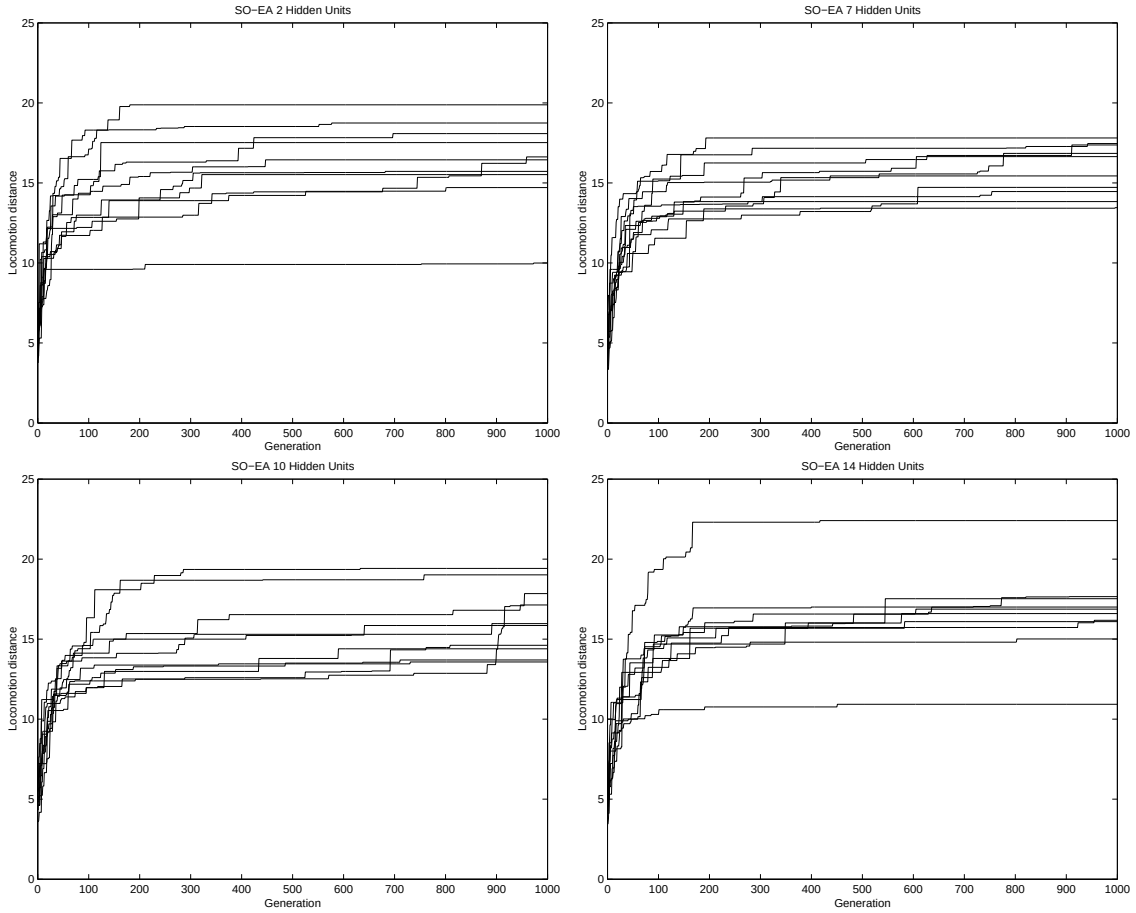

Figure 6.15: Best fitness for solutions obtained over 1000 generations for 10 runs using the single-objective EA with 1. 2 hidden units (top left), 2. 7 hidden units (top right), 3. 10 hidden units (bottom left), 4. 14 hidden units (bottom right). X-axis: Generation, Y-axis: Locomotion distance. Additional graphs can be found in the accompanying CD-ROM.

The evolution of the best solution using the single-objective EA for 10 runs over 1000 generations is shown in Figure 6.15 for four of the sixteen different hidden layer setups. Across all the different sizes of hidden layer used in the ANN controller, the majority of the improvement achieved in the best solutions occurred very early during evolution, as in the weighted sum method. This phenomenon could be seen to occur as early as the 50–80th generation in some of these runs. This can be explained by the fact that all the optimization effort is being focused solely on

a single objective and thus the single-objective EA should show a faster rate of improvement and subsequently be able to converge earlier compared to algorithms with distinct multiple objectives. However, the cost of this type of fast convergence is as discussed earlier in the previous paragraph, that the standard deviations between the best evolved solutions can be quite large. Figure 6.15.4 depicts this phenomenon clearly and is representative of the different hidden layer sizes used in the other runs for this single-objective EA. This shows that although very good solutions can be obtained in terms of locomotion distance, correspondingly poor solutions can also be expected.

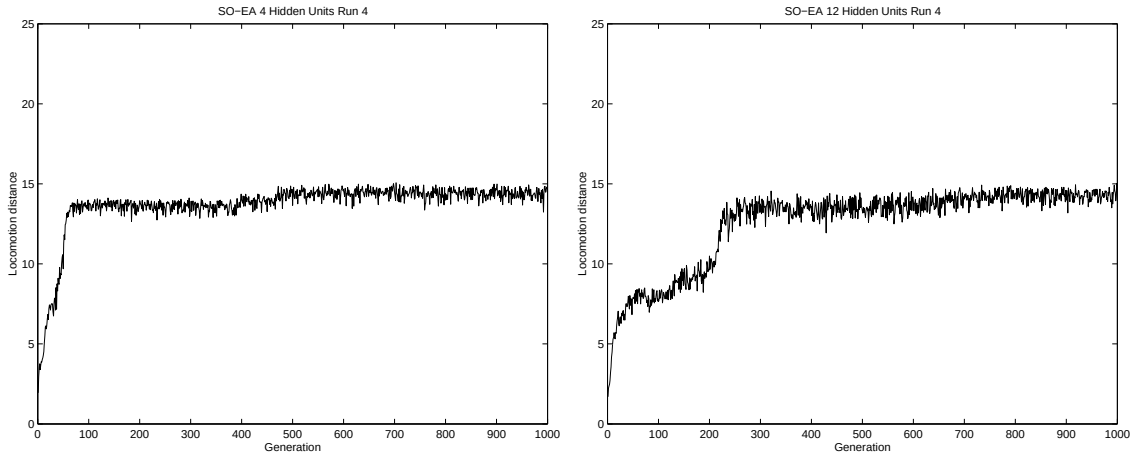

Figure 6.16: Mean fitness of population over 1000 generations using the fourth seed for the single-objective EA algorithm with 1. 4 hidden units (left), 2. 12 hidden units (right). X-axis: Generation, Y-Axis: Locomotion distance. Additional graphs can be found in the accompanying CD-ROM.

The mean locomotion distance and standard deviation for locomotion distance of the population as it evolved over 1000 generations using the single-objective EA is illustrated in Figures 6.16 and 6.17 respectively. The movement of the population means and standard deviations were very similar to the trends observed in the weighted sum EMO algorithm. The mean could be seen to increase and then either remain fixed or increase slightly over time (Figures 6.16.1 & 6.16.2). The mean also did not show any significant decrease in fitness throughout evolution.

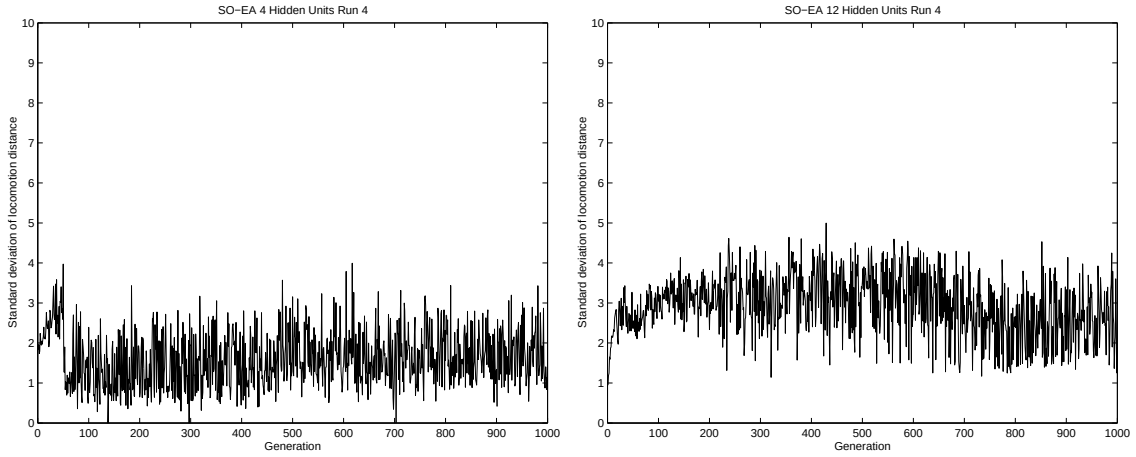

Figure 6.17: Standard deviation for fitness of population over 1000 generations using the fourth seed for the single-objective EA algorithm with 1. 4 hidden units (left), 2. 12 hidden units (right). X-axis: Generation, Y-Axis: Standard deviation of locomotion distance. Additional graphs can be found in the accompanying CD-ROM.

Likewise, the standard deviation movements (Figures 6.17.1 & 6.17.2) were again noticeably larger than those observed using the hand-tuned EMO algorithm. The highly similar trends observed in this single-objective EA and the weighted sum EMO algorithm strongly suggest that the use of strong elitism in both these algorithms is the likely cause of these phenomena. The effect of different sizes of hidden layers used in the controller did not appear to create any consistently different trends in the behavior of the population mean over time. However, the standard deviation of the population appeared to have varied within a lower range of deviation for the controllers that used less hidden units compared to those that used more hidden units. This observation suggests that although the population means were on average quite similar across different sizes of hidden layer used, the discrepancy between the best solutions and the newly generated solutions were larger in controllers that used more hidden units. Therefore, the analysis of the populations' average fitness and standard deviations points to the fact that the fitness landscapes may be more rugged for larger hidden layer sizes since newly produced offspring are further away

from the elite parents compared to controllers with smaller hidden layer sizes.

6.3.3 Search Space Characterization

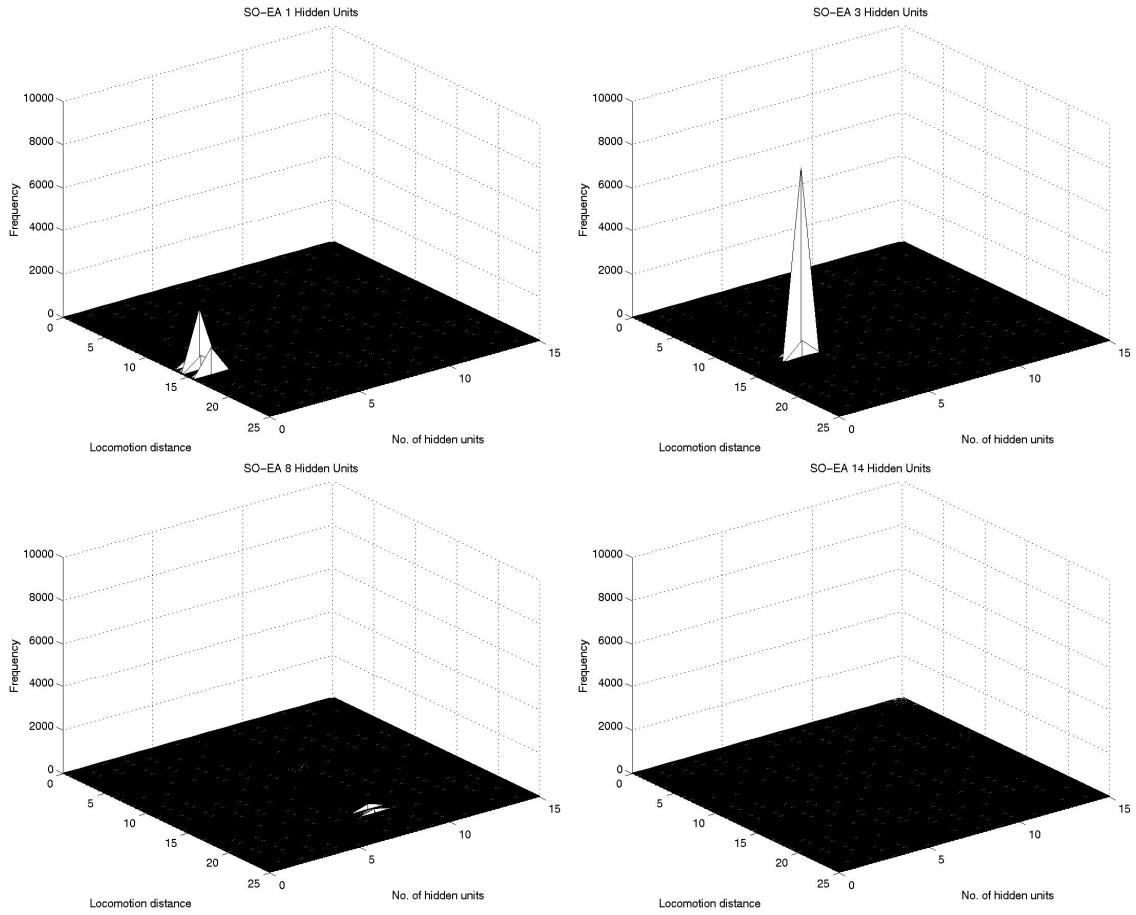

Figure 6.18: Frequency distribution of solutions obtained using the single-objective EA with 1. 1 hidden unit (top left), 2. 3 hidden units (top right), 3. 8 hidden units (bottom left), 4. 14 hidden units (bottom right). X-axis: Locomotion distance, Y-axis: No. of hidden units, Z-axis: Frequency. Additional graphs can be found in the accompanying CD-ROM.

The distribution of genotypes generated using the single-objective EA is plotted in Figure 6.18 in terms of locomotion distance and number of hidden units used in the ANN. Note that the frequency axis was again expanded to 10000 as in the analysis of the weighted sum approach (Section 6.2.3) to cater for the higher con-

centrations of solutions. The characteristics of the genotype distribution in the objectives spaces were expectedly very different from all prior algorithms since the size of the hidden layer was forcibly maintained within each evolutionary optimization setup by virtue of the single-objective methodology. Two different trends emerged from these runs. The distribution of solutions was more highly clustered when smaller hidden layer sizes were used, especially in setups that used between 0 and 4 hidden units. The distribution of genotypes depicted in Figures 6.18.1 and 6.18.2 are representative of this first trend. As the size of the hidden layer was increased, the generated genotypes were less closely concentrated, large peaks in the frequency were less commonly seen, and the magnitude of areas with high concentrations was also lower. The distribution of genotypes depicted in Figures 6.18.3 and 6.18.4 are representative of this second trend. Frequency distribution graphs at higher resolutions are shown below for the latter two hidden layer sizes.

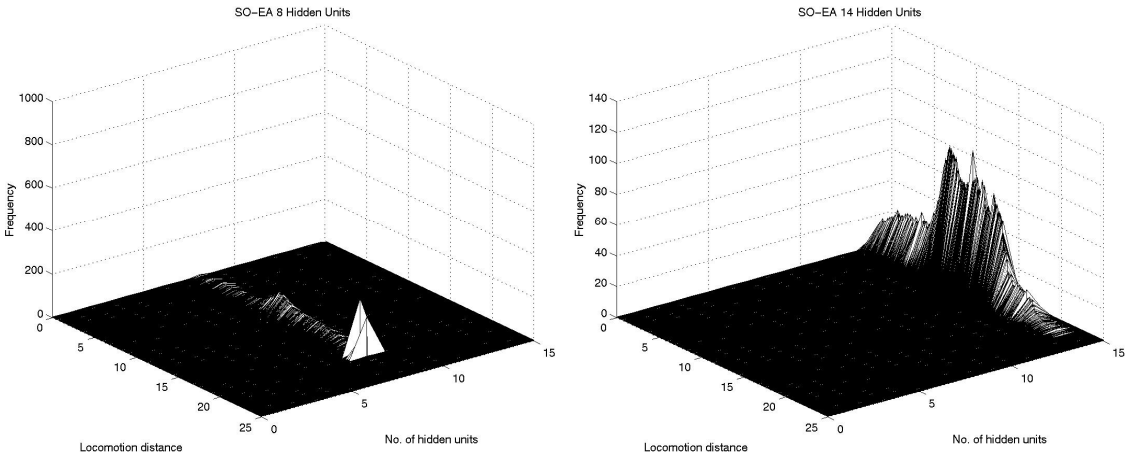

Figure 6.19: Frequency distribution of solutions obtained using the single-objective EA at higher resolutions for 1. 8 hidden units with frequency axis re-scaled to 1000 (left), 2. 14 hidden units with frequency axis re-scaled to 140 (right). X-axis: Locomotion distance, Y-axis: No. of hidden units, Z-axis: Frequency. Additional graphs can be found in the accompanying CD-ROM.

The distribution of controllers generated using 8 and 14 hidden units are illustrated in Figure 6.19 at increased resolutions to show the finer characteristics

of these fitness landscapes. Figure 6.19.1 is re-scaled to a frequency of 1000 and Figure 6.19.2 is re-scaled to a frequency of 140. Some clustering of genotypes can still be observed in controllers that used 8 hidden units. However, in controllers that used 14 hidden units, the distribution of genotypes was more uniformly distributed across the objective space. This phenomenon can be further explained by analyzing the associated contour graphs of these distributions.

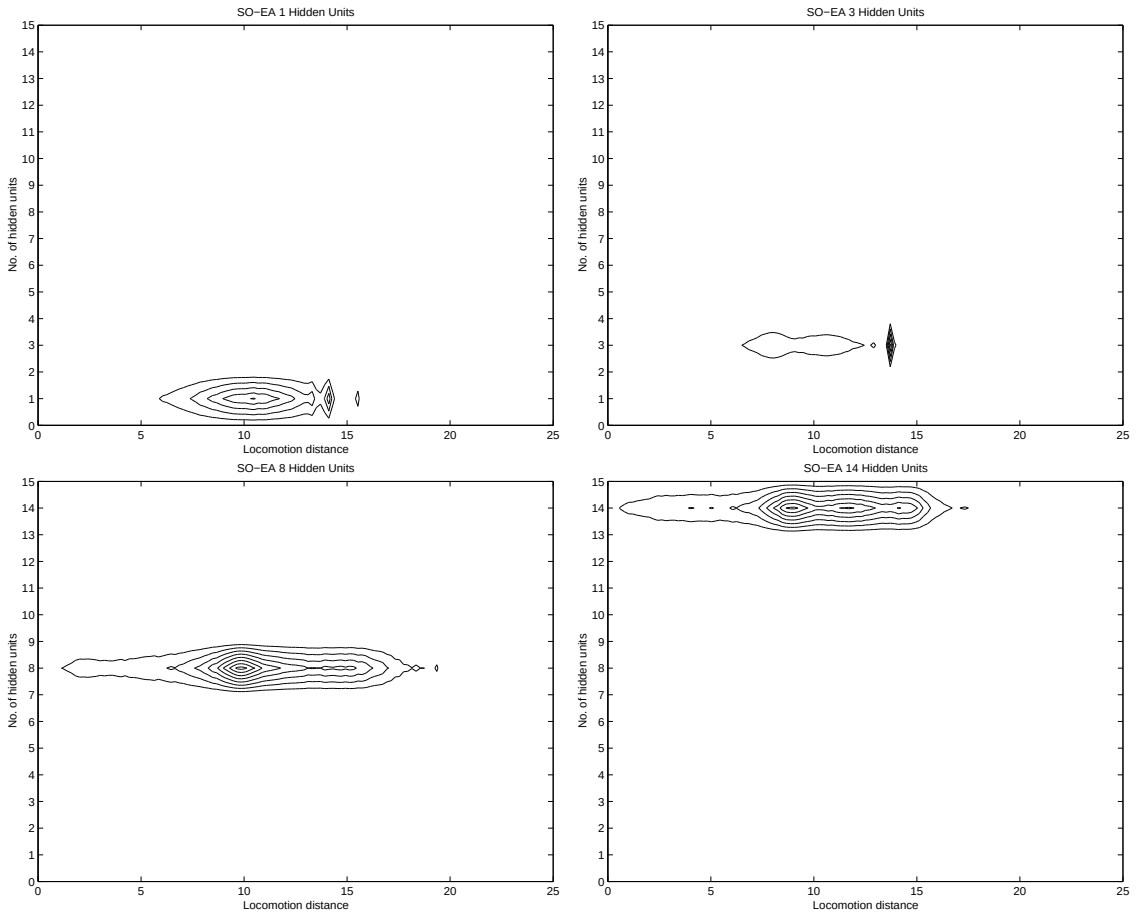

Figure 6.20: Contour graphs of frequency distribution of solutions obtained using the single-objective EA with 1. 1 hidden unit (top left), 2. 3 hidden units (top right), 3. 8 hidden units (bottom left), 4. 14 hidden units (bottom right). X-axis: Locomotion distance, Y-axis: No. of hidden units. Additional graphs can be found in the accompanying CD-ROM.

The contour graphs in Figure 6.20 illustrate the distribution of solutions

across the two objectives of minimizing hidden layer size and maximizing locomotion distance. The two trends that emerged in the 3D graphs discussed in the previous paragraph can be seen again clearly in these contour graphs. With a smaller number of hidden units, the solutions were distributed within a much smaller range of locomotion values, where most controllers achieved locomotion distances of between 6 and 14 units (Figures 6.20.1 & 6.20.2). Using a larger number of hidden units, the genotypes generated were able to sample larger areas of the objective space, where a significant proportion of the controllers were able to achieve locomotion distances that ranged between 1 and 17 units (Figures 6.20.3 & 6.20.4). This suggests that as the size of the hidden layer increases, the artificial creature is able to generate a wider range of locomotion capabilities in terms of the overall distance moved as a result of being able to sample the much larger search space offered by the increased network sizes.

The probability density function of solutions obtained using the single-objective EA is illustrated in Figure 6.21. Again, these graphs show that the smaller hidden layer sizes produced lower quality controllers in terms of locomotion distance compared to larger hidden layer sizes. Four hidden layer sizes of 8, 11, 12 and 14 were able to sample controllers up to 19 units of distance in the class of larger-sized networks before the probability dropped to 0. The graphs depicted in Figures 6.21.3 and 6.21.4 are indicative of the probability density functions obtained using these setups with larger hidden layer sizes. In comparison, for seven out of the eight hidden layer sizes that used 7 nodes or less, the probability of obtaining controllers dropped to 0 only around distances of between 15 and 17 units, as shown by the cumulative curves. The graphs depicted in Figures 6.21.1 and 6.21.2 are indicative of the probability density functions obtained using these setups with smaller hidden layer sizes.

In general, the search space characteristics were indicative of the fact that genotypes generated using smaller-sized hidden layers produced locomotion capabilities that were more constrained in terms of the range of distances achieved by the creature compared to genotypes that used larger-sized hidden layers. In the latter

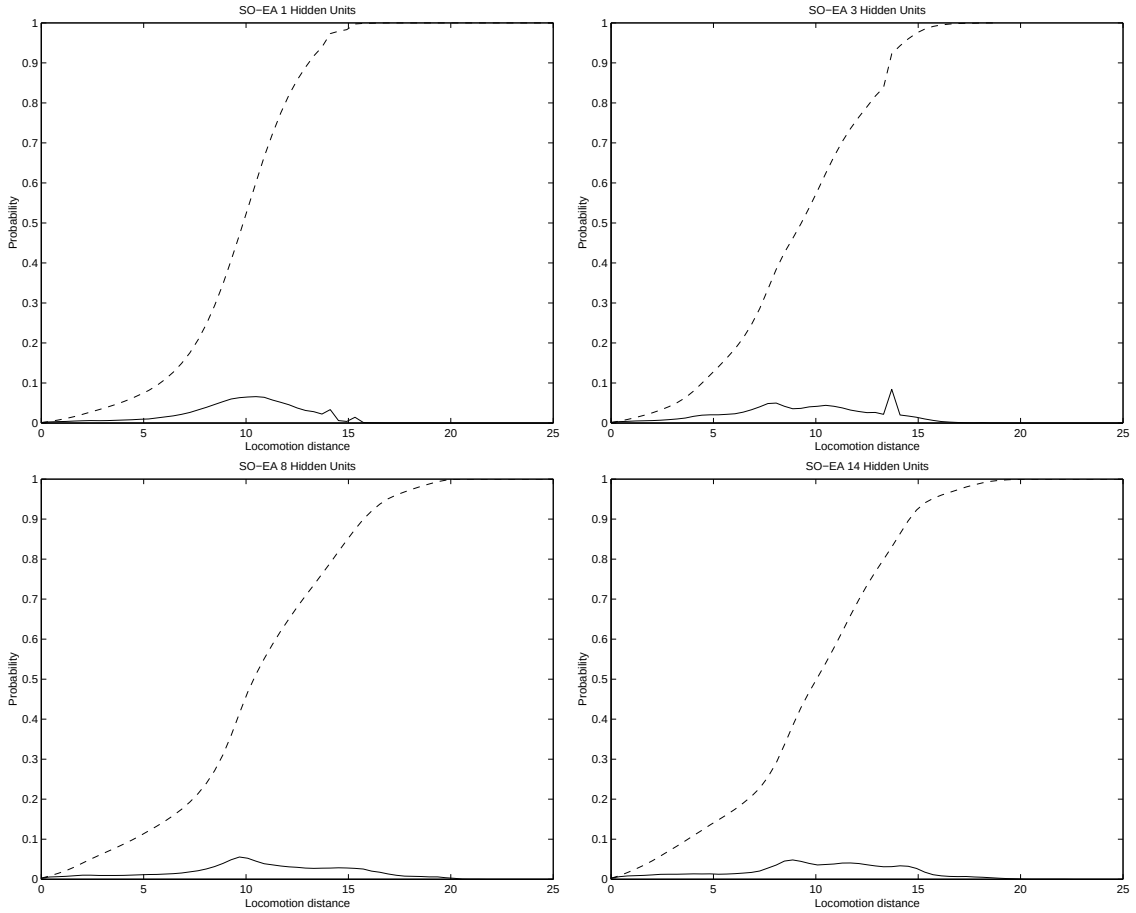

Figure 6.21: Density (solid) and cumulative (dashed) probability distribution of solutions obtained using the single-objective EA with 1. 1 hidden unit (top left), 2. 3 hidden units (top right), 3. 8 hidden units (bottom left), 4. 14 hidden units (bottom right). X-axis: Locomotion distance, Y-axis: Probability. Additional graphs can be found in the accompanying CD-ROM.

group of controllers, a significantly larger proportion of the newly generated genotypes were sampled from higher fitness sub-spaces. However, since there were no observable trends in terms of the best controllers found (see Table 6.3), these results strongly suggest that the underlying fitness landscape becomes increasingly rugged as the hidden layer size increases and coupled with the much larger search spaces, much more optimization effort may be required to find increasingly fitter solutions in these larger networks. This is an expected outcome given that the VC-dimension

increases with an increase in the number of hidden units (as explained earlier in Section 1.3) — therefore, more optimization effort is required.

6.4 Comparing SPANN Against Conventional Evolutionary Optimization Approaches

In this section, we compare the results obtained using the three EAs presented in this chapter against the results obtained using the SPANN algorithm. In the first section, SPANN is compared against the hand-tuned method, followed by a comparison against the weighted sum method in the second section, and then against the single-objective method in the third section. In the next two sections, firstly an overall discussion of the global Pareto solutions using SPANN as well as all other algorithms employed thus far is given in terms of the trade-off between the quality of the locomotion controller generated against the computational cost involved in obtaining these controllers, which is followed by an investigation into the amount of redundancy that is present in the overall best controllers evolved for locomotion distance using SPANN against the hand-tuned, weighted sum and single-objective methodologies. Finally, SPANN is compared to a well-known, state-of-the-art Pareto EMO algorithm called the Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb, Agrawal, Pratab, and Meyarivan 2000).

6.4.1 SPANN Against a Hand-Tuned EMO Algorithm

The advantage of using the self-adaptive Pareto approach against hand-tuning of crossover and mutation rates of the EMO algorithm is that it reduces the number of repeated experiments required to find the "right" combination of these parameters in order to generate the best possible solutions. Furthermore, it is unclear what effects fixing the crossover and mutation rates throughout the evolutionary optimization process will have on the quality of the eventual solutions obtained. In the results below, we compare the best solutions obtained from using 9 different combinations of crossover and mutation rates using the hand-tuned version

of the EMO algorithm against the best solutions obtained using the self-adaptive SPANN EMO algorithm.

Algorithm	Average Best Locomotion Distance ± Standard Deviation	t-statistic (against SPANN)	No. of Hidden Units
SPANN	13.9626 ± 1.7033	-	4.9 ± 2.6
HT-EMO **c**=10% **m**=10%	13.5192 ± 2.7845	(0.48)	3.2 ± 1.8
HT-EMO **c**=10% **m**=50%	14.1158 ± 2.6535	0.16	6.6 ± 2.1
HT-EMO **c**=10% **m**=90%	13.2843 ± 1.8225	(0.94)	8.1 ± 2.7
HT-EMO **c**=50% **m**=10%	14.1268 ± 2.1286	0.16	3.0 ± 2.4
HT-EMO **c**=50% **m**=50%	15.3819 ± 2.3195	1.39	6.5 ± 1.7
HT-EMO **c**=50% **m**=90%	13.1881 ± 1.4715	(1.03)	7.7 ± 2.5
HT-EMO **c**=90% **m**=10%	14.1511 ± 2.4721	0.20	3.4 ± 1.8
HT-EMO **c**=90% **m**=50%	13.1978 ± 1.6447	(1.42)	6.1 ± 1.4
HT-EMO **c**=90% **m**=90%	12.1653 ± 2.3799	(1.61)	6.8 ± 2.9

Table 6.4: Comparison of best locomotion distance for Pareto solutions obtained over 10 independent runs using the SPANN and hand-tuned EMO (HT-EMO) algorithms. **c** = crossover rate, **m** = mutation rate.

As shown in Table 6.4, the use of hand-tuned crossover and mutation rates did not provide any significant advantage over the SPANN algorithm in terms of the average best locomotion distance achieved by the evolved controllers. A t-test showed no significant differences at the $\alpha = 0.05$ and $\alpha = 0.01$ significance levels. Four combinations of the hand-tuned EMO algorithm gave marginally better results over the 10 runs in terms of the average best solution obtained while five other combinations performed worse than the SPANN algorithm. In terms of the number of hidden units used, three combinations in the hand-tuned EMO algorithm used an average of 1.7 nodes less than SPANN while six other combinations used an average of 7.0 nodes more than the SPANN algorithm. As such, the self-adaptive SPANN algorithm is beneficial compared to a hand-tuned EMO algorithm in that it reduces the computational runs required while still being able to maintain the same quality of solutions generated.

6.4.2 SPANN Against a Weighted Sum EMO Algorithm

A weighted sum approach combines the two objectives into a single objective by taking a weighted sum of the objectives. There are four main advantages of using the Pareto approach over a weighted sum method:

- The weighted sum method would only be able to generate a single Pareto solution in a single run compared to an entire set of Pareto solutions in a single run using the Pareto approach. Multiple runs will be required to generate a Pareto-front when using the weighted sum method.

- The determination of the weights is arbitrary in a weighted sum method. Some form of hand-tuning these weights will need to be carried out in order to obtain good results and as such, extra runs will again be required compared to the Pareto approach.

- The different objectives combined in a weighted sum method are assumed to be somehow commensurable, that is the objectives can be measured in the same units. In the case where they are not, as in this case of combining locomotion distance and number of hidden units, the use of the correct relative weights will be necessary to overcome this problem. Again, the Pareto approach does not require any such assumption to hold true since it treats each objective independently from the other.

- The weighted sum method assumes that the Pareto-front of the multi-objective optimization problem is of a convex nature. If the Pareto-front of the multi-objective optimization problem is actually non-convex, then the Pareto solutions generated by the weighted sum method will result in a discontinuous Pareto-front since the single-objective hyperplane will not be able to sample the non-convex regions of the Pareto-front. As such, in order to use a weighted sum method, the experimenter will first need to ascertain whether the particular problem is convex or otherwise, and no such information is usually available until the actual experiments are carried out and the Pareto-front plotted. Conversely, knowledge of such properties about the multi-objective optimization

problem is not required since the solutions generated using a Pareto approach is not constrained or limited to a specific Pareto-front of the multi-objective optimization problem.

Algorithm	Average Best Locomotion Distance ± Standard Deviation	t-statistic (against SPANN)	No. of Hidden Units
SPANN	13.9626 ± 1.7033	-	4.9 ± 2.6
WS-EMO γ=10%	9.8571 ± 1.4277	(5.97)	0.0 ± 0.0
WS-EMO γ=20%	10.4613 ± 2.6883	(3.19)	0.1 ± 0.3
WS-EMO γ=30%	8.4306 ± 1.3288	(7.96)	0.1 ± 0.3
WS-EMO γ=40%	9.4011 ± 2.1017	(4.46)	0.5 ± 0.8
WS-EMO γ=50%	11.3924 ± 3.0330	(2.15)	0.8 ± 0.9
WS-EMO γ=60%	12.1794 ± 2.9865	(1.86)	1.6 ± 1.0
WS-EMO γ=70%	13.7448 ± 2.4376	(0.33)	3.3 ± 1.9
WS-EMO γ=80%	14.0521 ± 2.3034	0.09	4.1 ± 1.5
WS-EMO γ=90%	15.1119 ± 1.9977	1.76	5.6 ± 1.9
WS-EMO γ=100%	15.2829 ± 3.6578	1.04	8.1 ± 1.5

Table 6.5: Comparison of best locomotion distance for Pareto/best solutions obtained over 10 independent runs using the SPANN and weighted sum EMO (WS-EMO) algorithms. γ = relative weight parameter.

In Table 6.5, we compare the weighted sum EMO against the Pareto SPANN algorithm. Results comparable to those obtained using the SPANN algorithm are achieved only with $\gamma = 70\%$. Although slightly higher locomotion distances were achieved using higher values of γ, which places more emphasis on the locomotion component of the weighted objective function, in all cases the standard deviation of the solutions were higher for the average best fitness for locomotion distance. Also, the case where $\gamma = 100\%$, which does not put any pressure whatsoever towards optimizing the size of the hidden layer, results in a very high average of hidden units used in the evolved controllers. This suggests that a significant amount of redundancy may be present in these networks, given that a t-test showed none of these weighted sum solutions were significantly better than those obtained with SPANN at both the $\alpha = 0.05$ and $\alpha = 0.01$ significance levels. Conversely, three of the weight combinations resulted in solutions significantly worse than SPANN at

the $\alpha = 0.01$ significance level ($\gamma = 10\%, 30\%, 40\%$) and one weight combination worse than SPANN at the $\alpha = 0.05$ significance level ($\gamma = 20\%$). Therefore, obtaining good solutions when using a weighted sum method critically depends on the choice of weights used on the respective objective functions and to find this right combination of weights would require multiple evolutionary runs to be conducted. Hence, the Pareto approach adopted in our SPANN algorithm is preferable from a computational cost point of view over a weighted sum method since it is able to proceed with the evolutionary optimization process without any tuning of weights and is still able to produce highly competitive results.

6.4.3 SPANN Against a Single-Objective EA

In a single-objective EA, the number of objectives that can be optimized in any one run is restricted to one. If there is more than one factor that may affect the quality of the solutions that are obtained from the evolutionary optimization process, then the single-objective EA will need to be re-run multiple times to test the effects of these other factors. For example, to test the effect of the size of the number of hidden units used on the evolution of artificial creature controllers, a separate set of runs will need to be carried out for each hidden layer size (see the experiments reported in Bongard and Pfeifer (2002) for such a case). As such, the obvious advantage an EMO algorithm has over a single-objective EA is its ability to optimize multiple objectives simultaneously, thereby significantly reducing the number of computational runs required to investigate other factors that may be crucial to the effectiveness of the evolutionary process.

In Table 6.6, we compare the single-objective EA against the multi-objective SPANN algorithm. In all the single-objective runs, higher locomotion distances were achieved by the evolved controllers in terms of the mean of the best solutions compared to SPANN. This is expected since all the evolutionary optimization pressure is focused only on the one objective of maximizing locomotion distance whereas this pressure is halved in the EMO case, where it is being shared with the objective of minimizing the hidden layer size. However, none of these results were significantly

Algorithm	No. of Hidden Units	Average Best Locomotion Distance ± Standard Deviation	t-statistic (against SPANN)
SPANN	4.9 ± 2.6	13.9626 ± 1.7033	-
SO-EA	0	15.7516 ± 2.9721	1.53
SO-EA	1	15.1441 ± 2.0260	1.33
SO-EA	2	16.3236 ± 2.7242	2.43
SO-EA	3	15.1532 ± 3.1696	1.05
SO-EA	4	15.2088 ± 2.2106	1.61
SO-EA	5	15.1562 ± 2.8741	1.32
SO-EA	6	16.0317 ± 2.0719	2.86
SO-EA	7	15.8033 ± 1.6159	2.07
SO-EA	8	17.4358 ± 3.2508	2.89
SO-EA	9	15.7375 ± 2.6430	1.53
SO-EA	10	16.1514 ± 2.1318	2.49
SO-EA	11	15.0614 ± 3.5612	0.86
SO-EA	12	15.4287 ± 3.0020	1.41
SO-EA	13	15.0359 ± 1.8909	1.19
SO-EA	14	16.6273 ± 2.8095	2.70
SO-EA	15	15.6150 ± 2.4605	2.58

Table 6.6: Comparison of best locomotion distance for Pareto/best solutions obtained over 10 independent runs using the SPANN algorithm and single-objective EA (SO-EA). Number of hidden units is fixed in the single-objective EA.

different at the $\alpha = 0.01$ significance level compared to SPANN while only six out of the sixteen different setups were significantly different at the $\alpha = 0.05$ significance level (number of hidden units = 2, 6, 8, 10, 14 & 15). However, it should be noted that the standard deviations in 15 out of the 16 different setups in the single-objective EA were higher than SPANN which suggests that even though the search space is much larger in the EMO case, the SPANN algorithm is still more stable in terms of its optimization results. It should also be remembered that 150 more evolutionary runs (15 setups $\times$ 10 repeats) were required in the single-objective case simply to investigate the effects of the hidden layer size on the evolution of these controllers. This would be a serious limitation for such investigations if the different number of setups required increases in magnitude (for example, consider the case where 100 or 1000 hidden units are allowed) or if the additional factors to be investigated are not discrete in nature (for example variations in the morpho-

logical parameters of the artificial creature). Moreover, in obtaining these better locomotion capabilities, there is a significant trade-off since the overall computational costs in terms of evaluating the ANN during evolution is much higher for the single-objective EA compared to SPANN (see Section 6.4.4 below).

6.4.4 Trading-Off Pareto Optimality Against Computational Cost

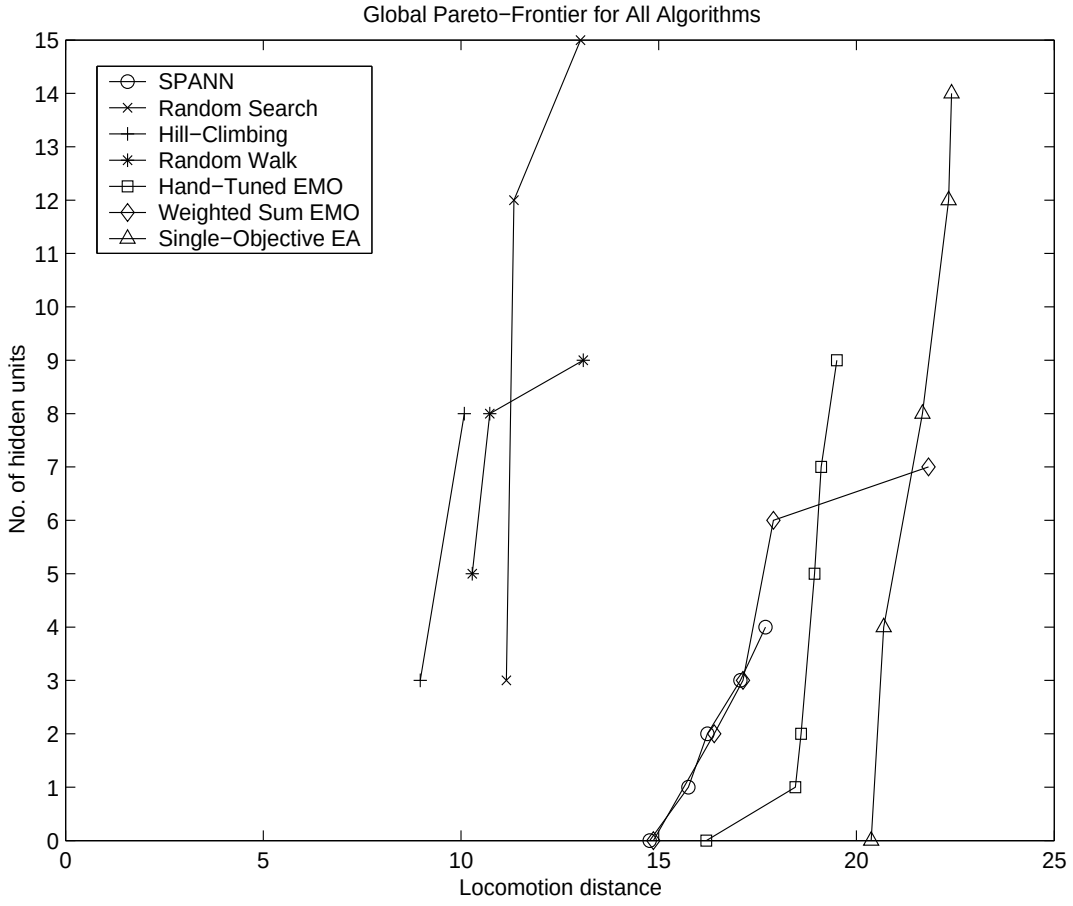

Figure 6.22: Pareto-front of solutions obtained using all algorithms over all runs conducted. X-axis: Locomotion distance, Y-axis: No. of hidden units.

Figure 6.22 plots the global Pareto-front for all the different algorithms used in evolving ANN controllers for the artificial creature. Two distinct groups of Pareto optimal solutions can be seen in this graph. One group is formed using the

random search, hill-climbing and random walk algorithms where the solutions were significantly worse than the other group of solutions formed by SPANN, the hand-tuned EMO, weighted sum EMO and single-objective EA. In the latter group, the best global Pareto-front was obtained using the single-objective EA when viewed in terms of the locomotion capability achieved, followed by the weighted sum method, the hand-tuned EMO and finally by SPANN. Although the Pareto-front of SPANN was not as optimal as the other algorithms, the controller with the highest locomotion distance discovered by each algorithm had the smallest hidden layer size for SPANN (4 nodes) compared to the hand-tuned EMO (9 nodes), weighted sum EMO (7 nodes) and the single-objective EA (14 nodes).

Algorithm	Best Locomotion Distance	$+/-\%$ of SPANN	Total Computational Cost	$+/-\%$ of SPANN
SPANN	17.6994	-	909,520,500	-
Random Search	13.0225	-26.4%	1,122,402,000	$+23.4\%$
Hill-Climbing	10.0832	-43.0%	1,047,833,000	$+15.2\%$
Random Walk	13.0900	-26.0%	1,124,060,000	$+23.5\%$
HT-EMO	19.5051	$+15.9\%$	7,529,814,400	$+727.9\%$
WS-EMO	21.8228	$+23.3\%$	3,073,867,500	$+238.0\%$
SO-EA	22.4069	$+26.5\%$	1,441,441,000,000	$+15748.4\%$

Table 6.7: Comparison of overall best locomotion controller obtained and corresponding computational cost using SPANN against all other algorithms.

Table 6.7 compares the overall best solution found by the different algorithms against the overall computation cost involved in discovering these solutions. The computational cost is estimated using the total number of hidden unit activations registered during the search process for each algorithm. This is a reasonable estimate since most of the computational time involved in conducting these experiments is spent on the evaluation of different ANN controllers by way of physically simulating the creature as guided by each newly generated controller within the Vortex physics-based world (see Section 3.1.1). Therefore, the 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 described by the following equation:

$$C = A \times G \times R \tag{6.4}$$

The best overall solution in terms of locomotion distance was obtained using the single-objective EA where the improvement over SPANN was 26.5%. However, the corresponding computational cost was a staggering 15,748% more than SPANN. This was mainly due to the number of repeated runs required for each different hidden layer size as well as the high usage of hidden units in the runs involving larger hidden layer sizes, which cannot be changed within each evolutionary run. Again, although better locomotion capabilities were obtained using the hand-tuned EMO and weighted sum EMO, dramatically higher computational costs were also associated with the use of the hand-tuned (727%) as well as the weighted sum methods (238%) compared to SPANN. Again these increases can be attributed to the need for repeated evolutionary runs required to test different weight assignments and crossover/mutation rates respectively in these algorithms. However, the inclusion of another objective in these two algorithms, which allowed for the minimization of the hidden layer size, did reduce the computational cost significantly compared to the single-objective EA. Although inferior results were obtained using the random search, hill-climbing and random walk algorithms, the overall computational costs were still higher than SPANN. This is due to the fact that these algorithms were only optimizing the locomotion component as in the single-objective EA and hence did not impose any pressure on minimizing the usage of hidden units during optimization.

In summary, although better controllers were evolved for locomotion distance using the single-objective EA, the corresponding trade-off in terms of overall computational cost was dramatically and unfavorably large. The trade-off between obtaining better locomotion capabilities and computational cost was again significantly and unfavorably large using the hand-tuned and weighted sum EMO algorithms. Hence, the SPANN algorithm has been shown to provide reasonably good results in terms of evolving locomotion controllers while at the same time providing the lowest overall computational cost compared to all other algorithms investigated

in this study.

6.4.5 Redundancy in Best Evolved Controllers

We now compare the amount of redundancy present in the overall best evolved controllers in terms of the hidden units as well as weight synapses obtained from SPANN against the hand-tuned, weighted sum and single-objective methodologies. For this set of experiments, we selected the overall best controller evolved for locomotion distance obtained from SPANN, the hand-tuned EMO algorithm, the weighted sum EMO algorithm and the single-objective EA as representative ANN architectures optimized using these respective approaches. These controllers, which are used in the lesioning experiments that test for redundancy in the ANN architecture, are listed in Table 6.8.

Algorithm	Locomotion Distance	No. of Hidden Units
SPANN	17.6994	4
HT-EMO	19.5051	9
WS-EMO	21.8228	7
SO-EA	22.4069	14

Table 6.8: Overall best locomotion controllers evolved using the SPANN, hand-tuned (HT-EMO), weighted sum (WS-EMO) and single-objective (SO-EA) algorithms used in the lesioning experiments.

In our analysis, redundancy is considered to be present when a controller can allow deletion of: (1) entire hidden units, or (2) individual weight synapses, without loss of fitness of more than 1 unit of locomotion distance compared to the intact controller originally evolved. The first comparison involved deletion of entire nodes in the hidden layer and can be regarded as macro-lesioning of the controller. This was done in a fashion similar to Migliono and Walker (2002) where all possible combinations of hidden units used in the ANN controller were systematically deleted. The lesioned controller is then re-evaluated and the new fitness achieved recorded. For example, if the best evolved controller had 4 hidden units, then all possible combinations of networks with 1 node lesioned are first evaluated, followed by all combinations of networks with 2 nodes lesioned and so forth terminating when

the next level of hidden unit removal causes the fitness evaluations of the lesioned controllers to fall below the redundancy threshold. The second comparison involved the deletion of individual weight synapses which can be regarded as micro-lesioning of the controller. The test for weight synapse redundancy was carried out in a greedy fashion due to the very large numbers of possible weight synapse combinations: first find the least loss of locomotion capability with 1 weight synapse lesioned, then proceed to find the next least loss of locomotion capability with another weight synapse lesioned by keeping the weight synapse found in the preceding step lesioned, and so forth terminating when the next level of weight synapse removal causes the fitness evaluations of the lesioned controllers to fall below the redundancy threshold. This second redundancy test is less drastic compared to the first test since the deletion of a single hidden node would cause entire sets of weight synapses to be also deleted in a single step. As such, the second redundancy test of lesioning only at the weight synapse level allows for a finer investigation into the controller's evolved architecture.

6.4.5.1 Hidden Unit Redundancy

Algorithm	Redundancy Fitness Threshold	Best Lesioned Fitness	Worst Lesioned Fitness	Average Lesioned Fitness ± Standard Deviation
SPANN	16.6994	12.8242	8.6991	10.9156 ± 1.9204
HT-EMO	18.5051	*18.6472*	1.6899	10.8691 ± 7.0303
WS-EMO	20.8228	18.6352	8.5114	14.4979 ± 3.8776
SO-EA	21.4069	17.5729	2.1421	12.9751 ± 4.8180

Table 6.9: Comparison of locomotion distance of overall best controller evolved using the SPANN, hand-tuned (HT-EMO), weighted sum (WS-EMO) and single-objective (SO-EA) algorithms with 1 hidden node lesioned.

None of the overall best controllers evolved for locomotion distance using SPANN, weighted sum and single-objective methodologies showed any redundancy in terms of hidden units present in the ANN controller. All possible combinations of controllers with a single hidden node removed from the optimized architecture using

these algorithms resulted in locomotion fitness below the redundancy threshold as shown in Table 6.9. However, the overall best controller evolved using the hand-tuned EMO algorithm did have one redundant hidden unit (the eighth node) which could be lesioned without causing the controller's capabilities to fall below the fitness threshold. This phenomenon together with the results from further levels of hidden unit lesioning of the overall best controller evolved using the hand-tuned EMO algorithm are discussed in the next paragraph. Surprisingly, the lesioning of a single hidden unit appeared to also have the most detrimental effect on the best controller evolved using the hand-tuned EMO algorithm in terms of the average loss of locomotion fitness compared to all other algorithms. Furthermore, the lesioning of a particular hidden node in the best controller evolved using the hand-tuned EMO algorithm also produced the worst 1-node lesioned controller, which only achieved a minuscule locomotion distance of just over 1.6 units. The removal of entire hidden nodes from the optimized ANN controllers seemed to result in large scale loss of locomotion capability suggesting that macro-lesioning of these evolved architectures is too drastic due to removal of not only a single hidden node but an entire set of weight synapses connecting to and originating from the lesioned hidden node. If the redundancy test were to be concluded at this coarse level, then the results would have indicated that no redundancy was present at all in the evolved controllers obtained using SPANN, the weighted sum EMO algorithm and the single-objective EA. However, a redundancy test at the finer weight synapse level, which is presented in the next section, showed otherwise.

Algorithm	Redundancy Fitness Threshold	Best Lesioned Fitness	Worst Lesioned Fitness	Average Lesioned Fitness ± Standard Deviation
HT-EMO	18.5051	18.1569	1.3326	9.4544 ± 5.6139

Table 6.10: Locomotion distance of overall best controller evolved using the hand-tuned EMO (HT-EMO) algorithm with 2 hidden nodes lesioned.

Before proceeding with the analysis at the weight synapse level, we first discuss the results obtained from the lesioning of different combinations of 2 hidden nodes from the overall best controller evolved using the hand-tuned EMO algorithm

since lesioning of 1 hidden node did show redundancy in this particular controller. None of the controllers which had 2 hidden nodes removed could produce locomotion fitness above the redundancy threshold as shown in Table 6.10. Compared to the overall best controller evolved using the self-adaptive SPANN algorithm, the hand-tuned EMO algorithm evolved a controller with more redundancy at the hidden unit level. Hence, the self-adaptive crossover and mutation rates appeared to have benefited the Pareto evolutionary optimization process in that SPANN was able to find a more compact network with less redundancy compared to the hand-tuned algorithm, which had pre-determined and fixed crossover and mutation rates during the optimization process.

6.4.5.2 Weight Synapse Redundancy

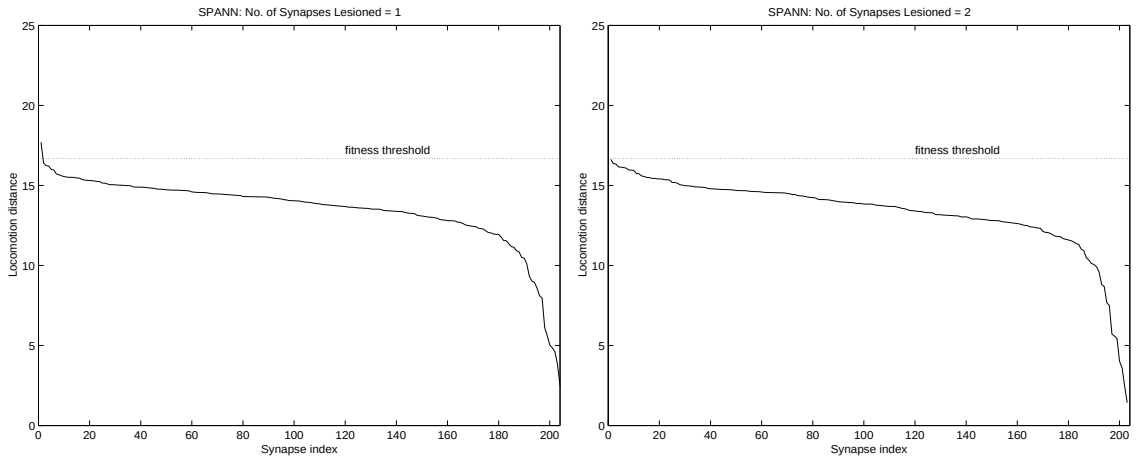

Figure 6.23: Locomotion distance for overall best controller evolved using SPANN with lesioning of 1. 1 weight synapse (left), 2. 2 weight synapses (right). The index assigned to the lesioned weight synapse is labelled according to the order of lesioning. X-axis: Synapse index, Y-axis: Locomotion distance.

Figure 6.23 illustrates the loss of locomotion fitness as weight synapses were lesioned in the overall best controller evolved using SPANN. There was only one particular weight synapse that could be lesioned without causing the controller's performance to fall below the fitness threshold (Figure 6.23.1). No other lesioning of

a single synapse could produce a fitness above the redundancy threshold. Additionally, no controller with 2 synapses lesioned could produce controllers that maintained their performance above the fitness threshold (Figure 6.23.2). This meant that the best evolved controller from SPANN only had a redundancy of one weight synapse and furthermore this occurred only with a specific weight synapse, which was the connection between the input from the joint sensor that measures the angle between the torso and the upper back left limb (Sensor x_1) and the first node in the hidden layer of the ANN.

Figure 6.24 illustrates the loss of locomotion fitness as weight synapses were lesioned in the overall best controller evolved using the hand-tuned EMO algorithm. This controller exhibited a high level of weight synapse redundancy where up to 281 synapses could be removed without causing the controller to fall below the fitness threshold. No controller with 282 synapses lesioned could produce controllers that maintained their performance above the fitness threshold (Figure 6.24.4). In line with results obtained from macro-lesioning at the hidden unit level, a much higher level of weight synapse redundancy should be expected in this controller compared to the overall best controllers evolved using the other algorithms. This is the case since the analysis from the previous section showed that an entire hidden node could be removed without causing the controller to fall below the fitness threshold, which correspondingly means that the entire set of synapses connected to and originating from this particular hidden unit were redundant. Thus, at least 30 weight synapses that are connected to this hidden unit can be removed without causing the lesioned controller to fall below the fitness threshold. The number of different weight synapses that could be removed varied considerably at different levels of lesioning. For example, Figure 6.24.2 showed that only approximately 25 different synapses could be removed at the 100-synapse level without causing the controller's locomotion capabilities to fall below the redundancy threshold. On the other hand, Figure 6.24.3 showed that approximately 75 different synapses could be removed at the 200-synapse level without causing the lesioned controller to fall below the fitness threshold. The fluctuations observed with regards to the number of different

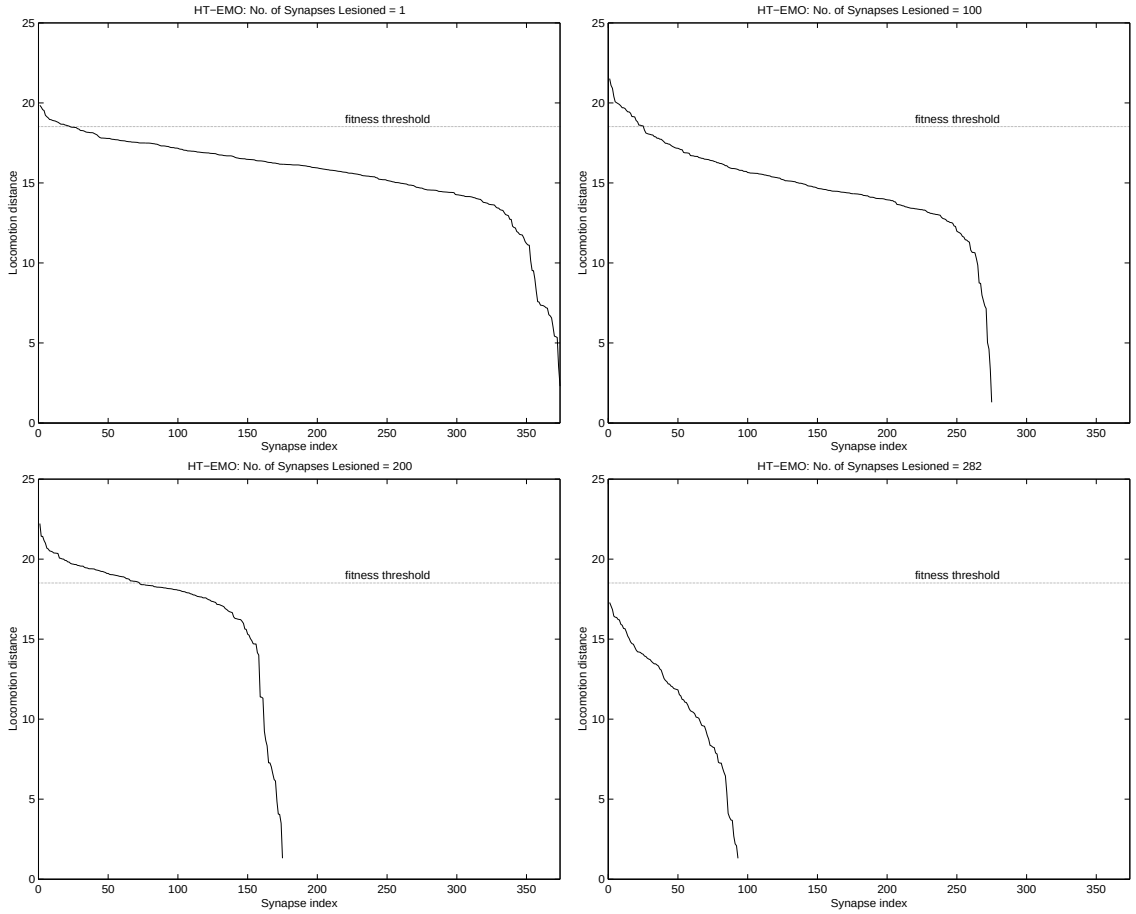

Figure 6.24: Locomotion distance for overall best controller evolved using the hand-tuned EMO algorithm (HT-EMO) with lesioning of 1. 1 weight synapse (top left), 2. 100 weight synapses (top right), 3. 200 synapses (bottom left), 4. 282 synapses (bottom right). The index assigned to the lesioned weight synapse is labelled according to the order of lesioning. X-axis: Synapse index, Y-axis: Locomotion distance.

synapses that could be removed at various levels of weight synapse lesioning are most probably due to the greedy nature in which the synapse lesioning takes place combined with the complex dynamics that takes place within the evolved ANN. A weight synapse lesioned presently will provide the best performance at the current level but the effects of this lesioning may at certain stages become highly sensitive to further lesioning and vice versa due to the combinatorial effects that occur between different weight synapses and hidden nodes in the network, which cannot be

ascertained by this one-step lookahead algorithm.

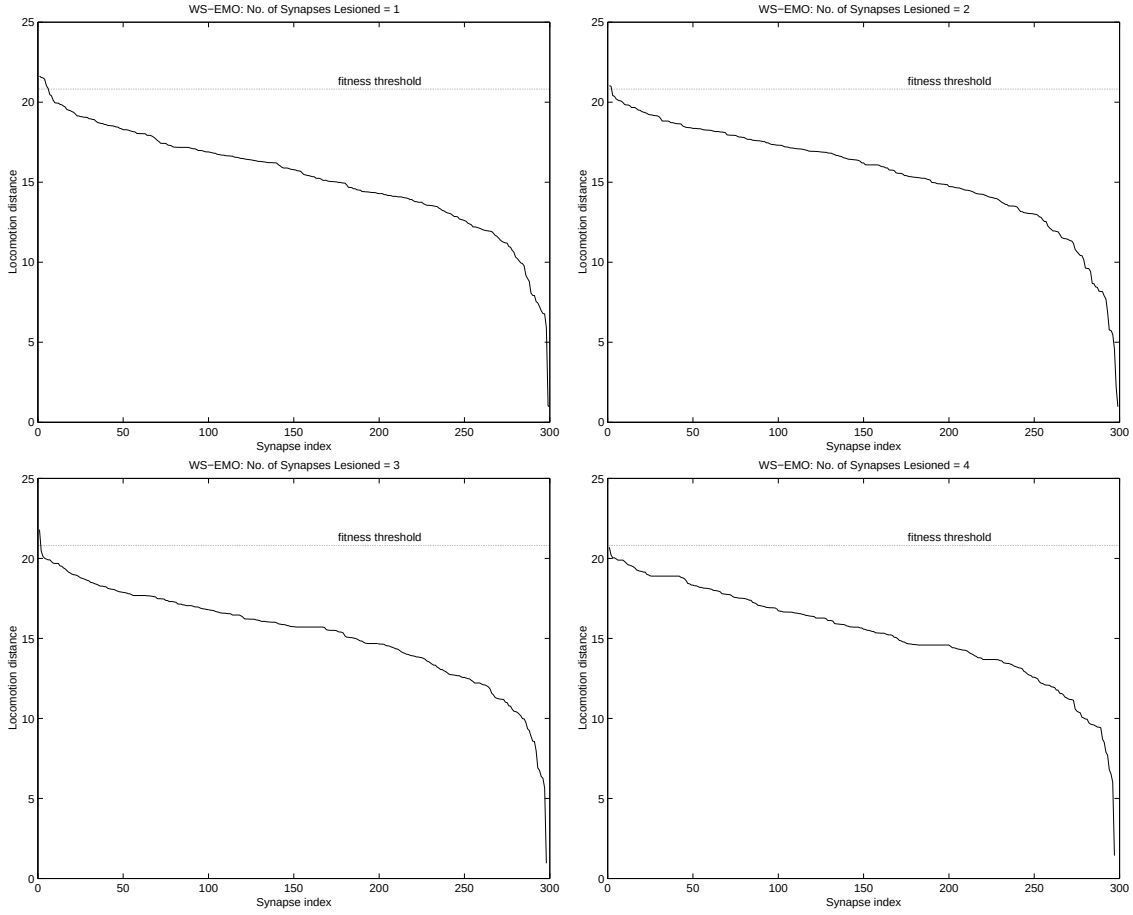

Figure 6.25: Locomotion distance for overall best controller evolved using the weighted sum EMO algorithm (WS-EMO) with lesioning of 1. 1 weight synapse (top left), 2. 2 weight synapses (top right), 3. 3 synapses (bottom left), 4. 4 synapses (bottom right). The index assigned to the lesioned weight synapse is labelled according to the order of lesioning. X-axis: Synapse index, Y-axis: Locomotion distance.

Figure 6.25 illustrates the loss of locomotion fitness as weight synapses were lesioned in the overall best controller evolved using the weighted sum EMO algorithm. Up to three synapses could be lesioned without causing the controller to fall below the fitness threshold. No controller with 4 synapses lesioned could produce controllers that maintained their performance above the fitness threshold (Figure 6.25.4). In terms of the number of different weight synapses that could

be removed and still produced controllers that performed above the threshold, 6 controllers were found when 1 synapse was lesioned (Figure 6.25.1), followed by 2 controllers when 2 synapses were lesioned (Figure 6.25.2) and finally by only 1 controller when 3 synapses were lesioned (Figure 6.25.3). Hence there was more synaptic redundancy present in the overall best controller evolved using the weighted sum method compared to SPANN.

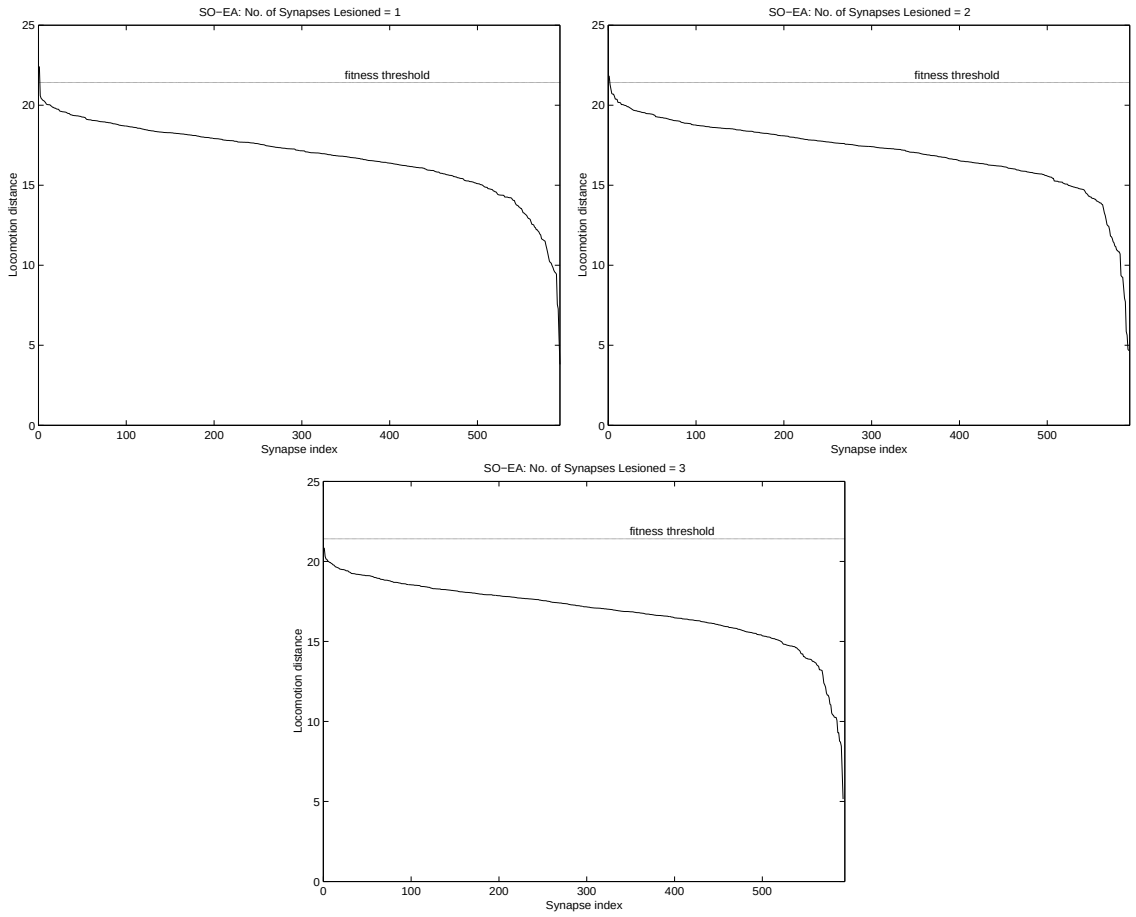

Figure 6.26: Locomotion distance for overall best controller evolved using the single-objective EA (SO-EA) with lesioning of 1. 1 weight synapse (top left), 2. 2 weight synapses (top right), 3. 3 synapses (bottom). The index assigned to the lesioned weight synapse is labelled according to the order of lesioning. X-axis: Synapse index, Y-axis: Locomotion distance.

Figure 6.26 illustrates the loss of locomotion fitness as weight synapses

were lesioned in the overall best controller evolved using the single-objective EA. Up to 2 synapses could be lesioned without causing the evolved controller to fall below the fitness threshold. No controller with 3 synapses lesioned could produce controllers that maintained their performance above the fitness threshold (Figure 6.26.3). Only a particular weight synapse could be removed when 1 synapse was lesioned which produced a controller that performed above the threshold (Figure 6.26.1). Following this lesioning, only another specific weight synapse could be removed at the 2-synapse lesioning level that resulted in a controller that performed above the threshold (Figure 6.26.2). Thus, there was also more synaptic redundancy present in the overall best controller evolved using the single-objective EA compared to SPANN but less synaptic redundancy compared to the hand-tuned and weighted sum method. A summary comparing the number of redundant weight synapses present in the best evolved controller obtained from SPANN against those obtained using the hand-tuned, weighted sum and single-objective algorithms is given in Table 6.11.

Algorithm	No. of Redundant Synapses
SPANN	1
HT-EMO	281
WS-EMO	3
SO-EA	2

Table 6.11: Number of redundant synapses in the best evolved controllers from the SPANN, hand-tuned (HT-EMO), weighted sum (WS-EMO) and single-objective (SO-EA) algorithms.

6.4.6 SPANN Against NSGA-II

To conclude our verification of SPANN as a beneficial Pareto EMO algorithm, we compare its results against one of the current state-of-the-art Pareto EMO algorithms called NSGA-II (Deb, Agrawal, Pratab, and Meyarivan 2000). Our objective here is simply to verify that the solutions obtained using SPANN are comparable to those obtained using a well-known and well-tested Pareto EMO algorithm

and not to determine which Pareto EMO algorithm is better for evolving locomotion controllers as this is beyond the scope of this thesis. The NSGA-II algorithm was obtained from the authors' web site (KanGAL 2003) and used as a benchmark algorithm without any modification to the NSGA-II algorithm. The NNType3 architecture was again used in this set of experiments and all other parameters remained the same: 1000 generations, 30 individuals, 500 timesteps and 10 repeated runs. NSGA-II requires a number of other parameters to be set by the user including the crossover and mutation rates which are non-self-adaptive. Recently, the authors of NSGA-II conducted a comprehensive comparative study of NSGA-II against other EMO algorithms, which were reported in Deb, Pratab, Agrawal, and Meyarivan (2002). Hence, in the first setup, these user-defined parameters were set according to those used in the above-mentioned comparative study as follows: crossover rate 90%, mutation rate for real-coded variables 0.1553% (representing the reciprocal of the number of real-coded variables), and mutation rate for binary-coded variables 6.6667% (representing the reciprocal of the number of binary-coded variables), distribution index for crossover operator 20, distribution index for mutation operator 20, and single-point crossover.

Algorithm	Overall Best Locomotion Distance	Average Best Locomotion Distance ± Standard Deviation	t-statistic (against SPANN)	No. of Hidden Units
SPANN	17.6994	13.9626 ± 1.7033	-	4.9 ± 2.6
NSGA-II Setup 1	15.5452	11.7421 ± 2.0497	(3.78)	0 ± 0
NSGA-II Setup 2	18.3941	16.2022 ± 1.5860	2.85	6.8 ± 2.3
NSGA-II Setup 3	20.4144	17.8635 ± 1.9744	4.54	8.4 ± 2.1
NSGA-II Setup 4	20.9806	16.2667 ± 2.1868	2.54	7.7 ± 1.7

Table 6.12: Comparison of best locomotion distance for Pareto solutions obtained over 10 independent runs using the SPANN and NSGA-II algorithms. Setup 1: c=90%, m(r)=0.1553%, m(b)=6.6667%. Setup 2: c=50%, m(r,b)=50%. Setup 3: c=50%, m(r,b)=90%. Setup 4: c=90%, m(r,b)=50%. **c** = crossover rate, **m(r)** = mutation rate for real-coded variables, **m(b)** = mutation rate for binary-coded variables.

Table 6.12 lists the best Pareto solutions for locomotion distance obtained using the NSGA-II algorithm and compares them against those obtained using the SPANN algorithm. The best solutions obtained using the first setup for NSGA-II produced controllers that used no hidden units in all 10 runs. The overall best locomotion distance achieved was lower than that obtained using SPANN. The very small mutation rate used in this setup most probably caused the evolutionary search to prematurely converge to local optima centered around controllers which did not use any hidden units. A t-test showed that the results obtained using NSGA-II were significantly worse than SPANN at the $\alpha = 0.01$ significance level for this setup. Also, the overall best controller from SPANN achieved over 2 units of distance more than the overall best controller obtained from this setup of NSGA-II (representing a decrease of 12.2% compared to the overall best locomotion distance achieved by SPANN).

To overcome the inferior results obtained using the setup reported in (Deb, Pratab, Agrawal, and Meyarivan 2002), a second experiment utilizing the best combination of crossover and mutation rates obtained from the hand-tuned EMO was conducted. This second setup used crossover and mutation rates of 50% with all other parameters unchanged. Much better results were obtained in this second setup, where the overall best controller in terms of locomotion achieved a higher distance than that obtained using SPANN by just under 0.7 units (representing a 3.9% improvement over the best locomotion distance achieved by SPANN). A t-test showed that the solutions obtained using NSGA-II with the second setup were significantly better than those obtained using SPANN at the $\alpha = 0.05$ significance level.

Since these results suggest that a high mutation rate may improve the performance of NSGA-II, we carried out a third experiment using a setup with an even higher mutation rate of 90% while maintaining the crossover rate at 50%. However, a t-test comparing the results from this third setup against the second setup for NSGA-II showed no significant improvements. To test whether a higher crossover rate would yield better results, a fourth experiment was conducted using

a setup with crossover rate of 90% and maintaining the mutation rate at 50%. Again, a t-test showed no significant improvements in the results obtained with the fourth setup compared to the second setup of NSGA-II. The solutions obtained with third and fourth setup for NSGA-II were significantly better than those obtained with SPANN at the $\alpha = 0.01$ and $\alpha = 0.05$ levels respectively. The best solutions obtained from the third setup of NSGA-II used an average of 8.4 hidden units, which is almost double the number used by the best solutions obtained using SPANN, while the fourth setup used an average of 7.7 hidden units.

The Pareto-frontiers obtained over the 10 runs of NSGA-II for the four setups are depicted in Figure 6.27. The first setup only produced controllers that did not make use of any hidden units in the ANN controller as can be seen in Figure 6.27.1. All runs converged to solutions that did not require any hidden layer transformation resulting in purely reactive controllers being generated. In comparison, the second, third and fourth setups which used much higher mutation rates all produced a much greater variety of controllers as shown by the Pareto-fronts plotted in Figures 6.27.2, 6.27.3 and 6.27.4 respectively.

Figure 6.28 plots the global Pareto-front of SPANN and NSGA-II. It can be seen that the Pareto-front generated through 10 runs of SPANN is comparable though dominated by the Pareto-front generated through 40 runs of NSGA-II (10 runs each in Setup 1–4). The solution with 0 hidden units of the NSGA-II global Pareto-front was contributed from the first setup of NSGA-II while the remaining 8 other solutions on the global Pareto-front were contributed from the other three setups.

In summary, as with the hand-tuned EMO algorithm, there is a trade-off between obtaining better locomotion controllers using NSGA-II at the cost of incurring greater computational expense to find the optimal parameter settings. It is clear that the performance of NSGA-II is also sensitive to the parameters used. As future work, it would be interesting to implement a self-adaptive version of NSGA-II for a direct comparison against the self-adaptive SPANN.

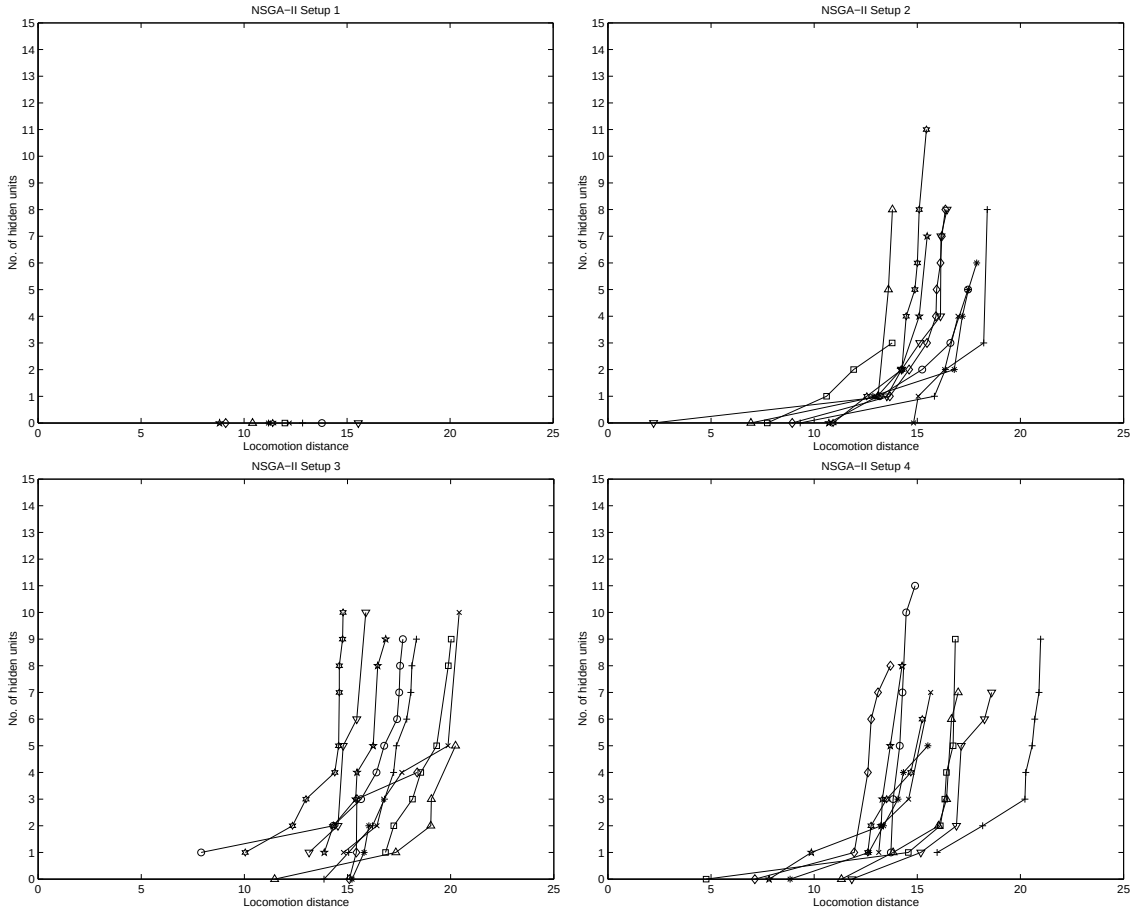

Figure 6.27: Pareto-front of solutions obtained for 10 runs using the NSGA-II algorithm for Setup 1 (top left), Setup 2 (top right), Setup 3 (bottom left), Setup 4 (bottom right). Setup 1: c=90%, m(r)=0.1553%, m(b)=6.6667%. Setup 2: c=50%, m(r,b)=50%. Setup 3: c=50%, m(r,b)=90%. Setup 4: c=90%, m(r,b)=50%. **c** = crossover rate, **m(r)** = mutation rate for real-coded variables, **m(b)** = mutation rate for binary-coded variables. X-axis: Locomotion distance, Y-axis: No. of hidden units.

6.5 Chapter Summary

The self-adaptive Pareto SPANN algorithm was compared against EMO algorithms that utilized hand-tuning of crossover/mutation rates and weighted sum approaches as well as a single-objective EA. It was found that SPANN discovered reasonably good quality controllers but most importantly required significantly less

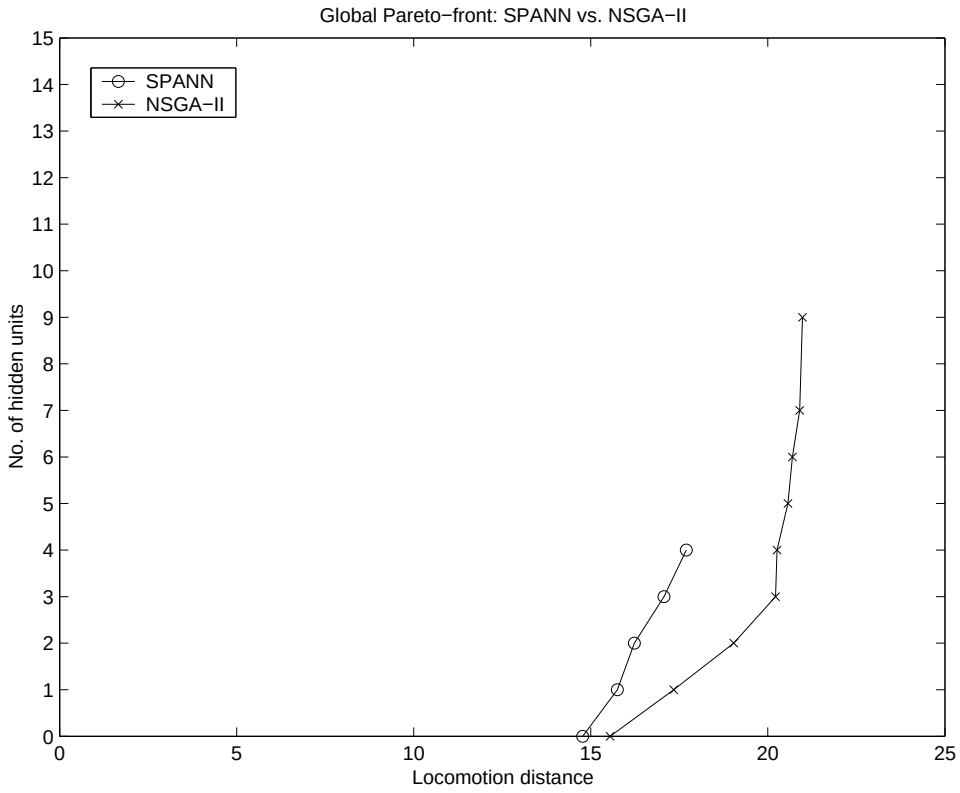

Figure 6.28: Global Pareto-front of controllers obtained using the SPANN and NSGA-II algorithms. X-axis: Locomotion distance, Y-axis: No. of hidden units.

overall computational costs. Although better solutions were found using the single-objective EA, the weighted sum and hand-tuned EMO algorithms, the trade-off in terms of computational costs was extremely high in comparison to SPANN. Furthermore, there was more redundancy present in the best controllers evolved using the hand-tuned, weighted sum and single-objective methodologies compared to the self-adaptive Pareto approach. The performance of SPANN was also found to be comparable to that of a current state-of-the-art benchmark Pareto EMO algorithm, NSGA-II. Therefore, the self-adaptive Pareto SPANN algorithm has been shown to be a highly beneficial EMO algorithm to use for evolving artificial creature controllers compared to other more conventional evolutionary optimization algorithms. In the next chapter, we will present a multi-objective view towards capturing and characterizing the complexities of evolved controllers using the SPANN algorithm.