To appear in *Proceedings of the 2002 Congress on Evolutionary Computation (CEC-2002), 2002.*

# Intelligent Process Control utilising Symbiotic Memetic Neuro-Evolution

A.v.E. Conradie*, R. Miikkulainen** & C. Aldrich*
*Department of Chemical Engineering, University of Stellenbosch, South Africa
**Department of Computer Sciences, The University of Texas at Austin, USA

**Abstract - A novel reinforcement learning algorithm, Symbiotic Memetic Neuro-Evolution (SMNE), is presented for neurocontroller development in non-linear processes. A highly non-linear bioreactor process is used in a learning efficiency case study. The use of implicit fitness sharing maintains genetic diversity and induces niching pressure, which enhances the synergetic effect between the global (symbiotic evolutionary algorithm) and the local (particle swarm optimisation) search. SMNE's synergetic effect accelerates learning, which translates to greater economic return for the process industries.**

## I. INTRODUCTION

Non-linear processes are prevalent in the chemical and metallurgical process industries. Non-linear dynamic characteristics may be intrinsic to the physics or chemistry of a process (i.e., distillation), or may arise in the coupling of simpler processes (i.e., heat exchangers). These dynamic characteristics are often complex and unpredictable, making autonomous control difficult to achieve. A controller's autonomy is reflected by its ability to maintain high and robust performance, despite an assortment of unexpected occurrences over a wide process operating range [1].

Linear controllers are often used in the process industries; despite the knowledge that linear controllers are unable to match the autonomy of rationally designed non-linear controllers in non-linear processes. Linear controllers thus incur an economic opportunity cost, as a consequence of sub-optimal performance in non-linear processes. Conversely, conventional non-linear control design necessitates extensive mathematical analysis, a significant degree of engineering judgement and expert process knowledge. Prior to commencing the non-linear controller design, the process engineer needs to have a clear understanding of how the control strategy will be implemented. Conventional non-linear control design methods are difficult to automate which hampers widespread implementation [2].

These difficulties can be surmounted by means of intelligent control techniques, such as reinforcement learning. Reinforcement learning is a computational framework, which requires minimal prior process knowledge for designing effective control strategies. However, reinforcement learning requires effective learning methodologies to extract control strategies from sparse learning information.

This paper introduces a novel learning methodology, Symbiotic Memetic Neuro-Evolution (SMNE), for developing neural network controllers in a reinforcement learning framework. A symbiotic genetic algorithm oversees the global search for an optimal neurocontrol strategy, while a Particle Swarm Optimisation (PSO) algorithm facilitates local search refinements to the neurocontrol strategy. This way, SMNE maintains a synergetic effect between the global and local search. Compared to other evolutionary approaches, this synergetic effect accelerates and improves the automated acquisition of process control knowledge from non-linear dynamic models.

## II. LEARNING METHODOLOGIES

### *A. Reinforcement learning*

Reinforcement learning is a computational framework that allows automation of the learning process. It is distinguished from supervised learning by not requiring exemplary training sets or even complete models of the dynamic environment. Reinforcement learning defines the interaction between a learning controller and its dynamic environment in terms of cause and effect information. Reinforcement learning therefore provides a means to program controllers without needing to specify how the control objectives should be reached. The control strategy is thus developed implicitly. This implicit learning characteristic sets reinforcement learning apart from conventional non-linear control techniques that require explicit process knowledge [3].

Reinforcement learning relies heavily on the learning methodology that uses controller performance evaluations to direct the learning process. Evolutionary algorithms (EA) are robust global optimisation methods for solving complicated combinatorial tasks, such as determining optimal controller parameters. Evolutionary algorithms have been used effectively as learning methodologies in reinforcement learning frameworks. In neurocontrol, evolutionary reinforcement learning searches in a population of possible neural network controllers for a strategy that encompasses effective control actions in the chemical process. Neurocontrollers are comprised of collections of neurons, with each neuron specifying the weights from the input layer (sensor readings) to output layer (control actions). In an EA framework, effective neurocontrollers produce offspring, which propagates effective neurons (genetic material) in the population. This genetic propagation of effective neuron structures is key to solving the combinatorial nature of neurocontroller parameter estimation [4].

### *B. Memetic algorithms*

EA's propagate effective neuron structures by varying the sample distribution in the solution space, depending upon the evaluation of the objective (fitness) function. This selection biases the search towards regions of the solution space where near optimal solutions have been discovered. Local refinements to these near optimal solutions could significantly accelerate arriving at an optimal solution. However, EA's are not suited to focusing local refinements in large combinatorial tasks. Genetic evolution may be augmented to facilitate local (neighbourhood) search via cultural evolution [5].

Analogous to genetic propagation, cultural transmission (i.e., bird song) is the evolutionary flow of information. However, there are significant differences between cultural and genetic evolution. In cultural evolution, improvements are seldom a result of copying errors or the exchange of co-adapted units of information. Clear-cut combination of exact ideas does not generally lead to innovation. An idea is rather blended with other similar ideas based upon perception and understanding. This blending process is the driving force towards innovation. Genetic evolution does not incorporate an innovative component, as experimentation (reproduction) with new information is governed by biased selection. A gene is not changed based on the quality of other similar genes. The individuals in cultural evolution are conscious entities that use one another's ideas in the search process, subject to cooperation and competition. Genetic evolution has no concern for individual genes, but focuses on improving the population by propagating effective gene combinations [5].

Memetic algorithms (MA) are evolutionary algorithms that use cultural evolution for local search (LS). The local search is applied to solutions in each generation of the EA, creating a process of lifetime learning. The EA searches globally for regions containing significant optima, while the LS searches these regions for the local optimum. The EA is thus responsible for exploration, whilst the LS governs exploitation. A balance between exploration and exploitation ensures that the minimum number of evaluations is employed in finding the global optimum. This balance is dependent on the synergy between lifetime learning and evolution [5].

LS aids the evolutionary process by smoothing the fitness landscape. LS exploits the local fitness landscape, which absolves the EA from devoting resources to searching in areas of local complexity on the fitness surface. This smoothing essentially involves a discretisation of the fitness landscape. Consider the optimisation of the fitness landscape in figure I.

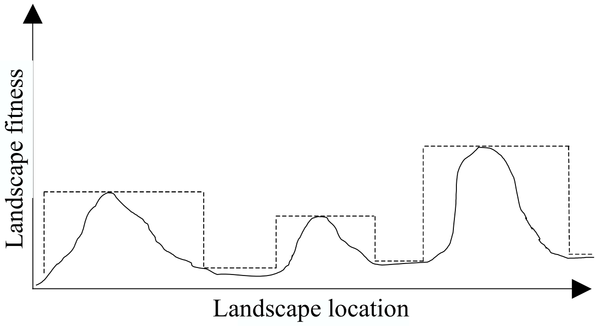

**Figure I - Smoothing of the fitness landscape by local search, thereby reducing the complexity of the EA's solution space.**

Assume that any EA solution, located on one of the slopes on the three peaks, is able to locate the local maximum through LS. The EA's task is simplified considerably, in that it only needs to locate three regions of the search space. The dashed lines in figure I indicate these three discrete regions. With the added local search capability, the complexity of the EA's solution space is reduced significantly. The plasticity afforded by lifetime learning makes it easier for the EA to climb to peaks in the fitness landscape [5].

Therefore, the EA of a memetic algorithm should not generate multiple solutions in the neighbourhood of a single optimum, but should maintain a diverse (wider) search in the solution space. Thereby, the EA aids the LS by bordering regions (sub-spaces) of the fitness landscape that contain significant optima. Such regions become prime candidates for exploitation by local search algorithms. A synergetic effect, which accelerates evolution, thus exists in an evolving population of individuals, where the individuals are also exposed to learning during their lifetime [5].

A key element to maintaining such synergy is a diversification mechanism in the EA. Genetic diversity is required to continue a global search. Global reliability, which promises convergence to the global optimum, is required to ensure that every region of the solution space is effectively explored [5].

## III. NOVEL COMBINATORIAL SEARCH

### *A. Effective genetic diversification*

Genetic diversity prevents convergence to a local optimum and allows continued genetic search, assuring global reliability. The continued introduction of informational variety is critical for effective exploitation by cultural evolution. Numerous methods for maintaining genetic diversity have been proposed; such as crowding, distributed sub-populations with migration, local mating and explicit fitness sharing. These methods are effective in slowing convergence, but have been unable to sustain a diverse dynamic equilibrium in the EA's population [1].

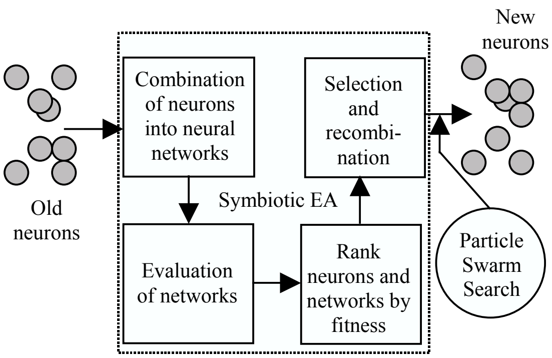
New neurons
Combination of neurons into neural networks
Selection and recombi-nation
Symbiotic EA
Old neurons
Evaluation of networks
Rank neurons and networks by fitness
Particle Swarm Search

**Figure II - Flow diagram for a single generation in the Symbiotic Memetic Neuro-Evolution algorithm.**

Implicit fitness sharing entails the search for partial solutions, which cooperate to encode the complete solution. In neuro-evolution, individual neurons are partial solutions to the complete solution (neural network). Neurons that compete to perform the same task, compete for the same rewards, namely weak cooperation. Neurons that do not overlap in their tasks, are cooperating in an indirect manner, namely strong cooperation. Strong cooperation is symbiotic in nature [6].

Strong cooperation maintains high quality diversity in the face of significant selection pressure, by balancing convergence with the restorative force of niching pressure. A niching phenomenon also implies several parallel searches for partial solutions, which should prove more effective than a single search for the complete solution [6].

*B. Symbiotic Memetic Neuro-Evolution*

Implicit fitness sharing and the synergetic effect of memetic algorithms, may be combined to enhance global reliability and accelerate evolution in complex combinatorial tasks. A novel memetic algorithm, Symbiotic Memetic Neuro-Evolution (SMNE), is introduced for developing neurocontrollers in a reinforcement learning framework. A symbiotic genetic algorithm (figure II) is employed to ensure global reliability, while performing an aggressive explorative search. Particle Swarm Optimisation (PSO), a cultural evolution method, is used for local exploitative search after each EA generation.

*1) Symbiotic evolutionary algorithm*

Similar to the SANE algorithm [4], the symbiotic EA (dashed box in figure II) maintains both a neuron and a network population. Each member of the neuron population encodes a hidden neuron, with weights from the input layer to the output layer. While SANE maintains a single neuron population, SMNE's neuron population is comprised of a number of sub-populations. The network population is constructed from the neuron population. Each network is a collection of pointers to the neuron sub-populations. Each position in a network's hidden layer is filled from a different neuron sub-population.

Competing conventions is avoided in the SMNE's network population, as each network position points to a particular sub-population. Competing conventions is also avoided in the neuron population, as each weight connects to a fixed input or output throughout evolution. Real-value encoding of neuron weights also ensures that the crossover location is at the gene (weight) boundaries. This causes less gene disruption during crossover. Unchanged genes are thus carried into the next generation, focusing the search.

Each network is evaluated in the reinforcement learning task (i.e., process control) and assigned a fitness value based on the control performance criteria. High network fitness reflects superior performance in the control task. The network population is ranked after evaluation. The neuron fitness assignment implements implicit fitness sharing. Each neuron is assigned a fitness value, based on the fitness values of the five best networks in which it participated [4]. High neuron fitness reflects a neuron's ability to cooperate with other neurons in different sub-populations. Rewarding neuron cooperation with high fitness values, induces niching pressure in the sub-populations. Strong cooperation between sub-populations facilitates the search for partial solutions that comprise the complete solution. Each sub-population thus serves as a container in which a niche may emerge. The niching pressure retains genetic diversity in the neuron population, allowing the genetic search to continue. The neuron population is ranked after evaluation. Recombination and reproduction is based on the network and neuron ranking [4].

One-point crossover is applied to the elite neuron population (top 20%). The elite neurons breed across the sub-populations, thereby exploring the solution space between current niches. Each elite neuron randomly selects (on rank) a mate that has a higher fitness than itself. Two effective parents should produce offspring with similar or superior performance. As the best elite neurons are more likely situated in different sub-populations (strong cooperation), their offspring attempt combining two functionalities into a single neuron. This may free sub-populations to pursue other niches. Crossover in the poorer elite neurons has a greater probability of selecting a parent from the same sub-population, which focuses the genetic search (weak cooperation). Each offspring neuron is copied to a neuron sub-population, depending on the gene contribution from each parent. The offspring neurons replace the worst neurons in each sub-population. Mutation is applied, with low probability (2%), to the remainder of the neuron population.

An elite network population (top 20%) retains knowledge of effective neuron combinations [4]. The elite network population's reproduction operator replaces a neuron pointer with one of its offspring that was copied to the same sub-population. This reproduction operator applies, with 25% probability, to all the neuron pointers in the elite network population. The offspring networks replace the worst networks. The remaining networks are constructed randomly from the sub-populations, with a propensity for selecting offspring neurons of the elite neuron population. This scheme ensures that neurons not selected in the previous generation, obtain pointers and therefore a fitness evaluation.

*2) Particle Swarm Optimisation*

A local refinement search, Particle Swarm Optimisation (PSO), augments SMNE's symbiotic EA. PSO is implemented after each EA generation, as illustrated in figure II, thereby inducing lifetime learning. PSO is a population based optimisation method loosely based on the social behaviour of flocks (swarms) of birds. PSO updates each individual based on the collective experience of the swarm individuals in the solution space [7]:

$$v_{id} :| \varpi \,|\, v_{id}\ 2\ c_1 \,|\, rand() \,|/\, p_{id}\ 4\ x_{id} \,\|\, 2$$
$$c_2 \,|\, rand() \,|/\, p_{gd}\ 4\ x_{id} \,\| \qquad (1)$$
$$x_{id} :| x_{id}\ 2\ v_{id} \qquad (2)$$

where $v_{id}$ is a particle's velocity vector, $\varpi$ is the inertia weight, $c_1$ and $c_2$ are constant parameters and $x_{id}$ is a particle's position vector. Each particle retains partial knowledge of its own best position, $p_{id}$, and the position of the best swarm particle, $p_{gd}$ (equation 1). Based on these two knowledge components, each particle updates its velocity vector to determine its next position (equation 2) [7]. PSO shares numerous characteristics with cultural algorithms. The swarm's movement in the solution space is akin to cultural transmission and the innovative blending of ideas. Also, each particle's momentum protects against entrapment in a local optimum.

Each particle thus blends its own experience and that of the best swarm particle in a unique manner. PSO's assumes that the best swarm particle is located in a region of the solution space that contributes to solving the control task. Each particle moves uniquely in the general direction of the best swarm particle. This may lead to the discovery of superior, adjacent regions of the solution space. A new best swarm particle consequently moves the swarm in a new direction. PSO thereby involves cooperation as a result of shared experience and competition for superior fitness [7].

SMNE's particle swarm implementation incorporates a small inertia weight ($\varpi$ = 0.4 in equation 1) to facilitate local search. The parameters $c_1$ and $c_2$ (equation 1) are also equal to 0.5 (conventionally 2.0), which ensures an exploitative search. Each neuron sub-population contains a separate PSO implementation. The PSO's neuron weight changes are Lamarckian, that is, the weight changes update the genes. The best neuron in a sub-population is the best particle in its swarm. PSO refines each partial solution search, by sharing the best neuron's control knowledge with other neurons in its sub-population.

Local search (LS) should only be applied to evolutionary solutions where it will be the most beneficial. A Lamarckian LS may also begin to dominate the genetic population, causing a loss in diversity. LS should thus only be applied to a sub-set of the total population [5]. Therefore, the neurons of the elite network population constitute the candidates for LS. These neurons are presumably located in close proximity to the regional sub-spaces that contribute to the search for partial solutions. Complete LS may also involve a large number of evaluations. The LS is only applied for the limited number of five steps. This ensures that evaluations are effectively utilised, after each evolutionary step. This partial local optimisation is in keeping with cultural evolution, ensuring that each particle modifies the swarm knowledge uniquely. Applying LS for a limited number of steps, also avoids a loss of diversity in the elite network population's neurons.

As discussed in section II, the symbiotic EA and the PSO of SMNE result in a synergetic search algorithm that allows effective discovery and refinement of partial solutions. The SMNE algorithm was subsequently applied to a challenging real world problem.

## IV. COMPARATIVE BIOREACTOR CASE STUDY

The SMNE algorithm was use to develop an optimal neurocontroller for a simulated bioreactor. Stochastic hill climbing, the SANE algorithm [4], SMNE's symbiotic EA and complete SMNE were tested in a comparative study of these learning methodologies.

### *A. Bioreactor benchmark*

A bioreactor is a continuous stirred tank fermenter. It contains a biomass of micro-organism that grow by consuming a nutrient substrate that is continuously fed into the reactor. The biomass is filtered and sold as the product. Fermentation control presents a challenging control problem, as the bioreactor's dynamic behaviour is invariably highly non-linear and model parameters vary in an unpredictable manner. The biochemical processes are complex and create a multitude of steady state attractors, which may be stable, chaotic or periodic depending on the residence time in the bioreactor. The process state is difficult to quantify, owing to unreliable biosensors and long sampling periods [2].

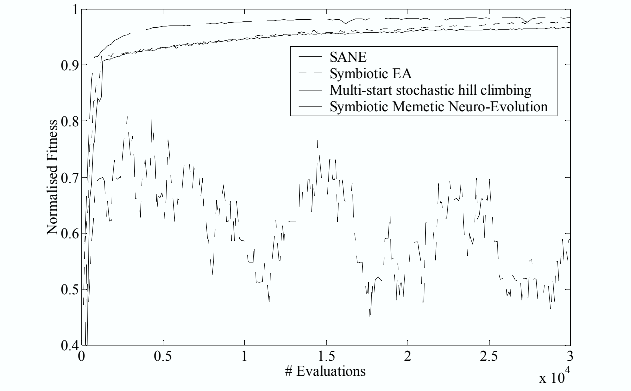

**Figure III - Average normalised venture profit for the learning methodologies.**

TABLE I
ANOVA ANALYSIS

| *ANOVA analysis* | *P-Value* |
|---|---|
| Stochastic hill climbing vs. SANE | $8.9 \cdot 10^{-6}$ |
| SANE vs. SMNE's symbiotic EA | $5.1 \cdot 10^{-4}$ |
| Symbiotic EA vs. SMNE | $2.1 \cdot 10^{-3}$ |

The control objective is to maximise the venture profit of the process. This entails locating the operating region with the maximum venture profit and ensuring effective control actions. The operating region with maximum venture profit requires stabilising a chaotic attractor. No process information is provided with regard to the execution of the control task. The bioreactor constitutes a significant real-world process control task for testing reinforcement learning methodologies [2].

### *B. Experimental set-up*

A comparative study was undertaken to test the contribution of the SMNE algorithm in developing neurocontrollers. SMNE was compared to three other methods: (1) multi-start stochastic hill climbing (SHC) as a reduced model, augmented with an initial random search for suitable initial starting points, (2) the SANE algorithm [4] as a symbiotic EA with a single population of neurons, and (3) the symbiotic sub-population EA used in SMNE without PSO. The neurocontrollers received the bioreactor's three sensor readings as inputs, and generated the positions of its four valves as outputs. Twenty learning simulations, each for a total of 30 000 evaluations, were run for each learning method. Each evaluation was initialised to a random process state and run for 300 sample periods. The fitness for each evaluation was calculated as:

$$f \mid \int_{0}^{300} t \lceil p(t) \lceil dt \qquad (3)$$

where *f* is the fitness value and *p(t)* is the instantaneous profit at sample period, *t*. The statistical significance of the performance differences was measured using ANOVA analyses.

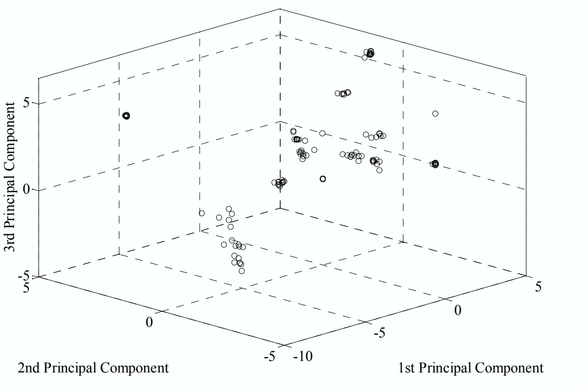

**Figure IV - Principle components of the weight vectors of each neuron weights in the elite network population.**

### *C. Results*

The average normalised venture profit for each method is shown in figure III. The ANOVA results are tabulated in table I. Stochastic hill climbing could not learn an effective control strategy for the bioreactor (figure III). SHC could not reliably progress beyond the initial basin of attraction for any of the twenty simulation runs. This demonstrates the complexity of the bioreactor's dynamics and justifies using more complex algorithms in solving the control task.

The EA algorithms proved more successful in progressing towards the global optimum (venture profit = 1). Although SANE and SMNE's symbiotic EA are similar in implementation, SMNE's symbiotic EA deals more effectively with competing conventions and focuses the niching phenomenon. The ANOVA analysis indicates that the sub-population treatment is statistically superior (ANOVA $p < 0.01$) to the single neuron population SANE algorithm.

SMNE, with its local search refinement operator (PSO), both accelerated and produced a higher average venture profit than the EA implementations alone (figure III). The ANOVA analysis (ANOVA $p < 0.01$) also indicates that the PSO treatment in SMNE is statistically superior to the EA implementations alone.

Figure IV presents a principal component analysis (first three principal components, explain 77% of the variance). Each marker represents a neuron in the elite network population. The key elements of SMNE are illustrated: (1) the observed clusters illustrate the niching pressure induced by implicit fitness sharing, (2) genetic diversity has been maintained, allowing continued exploratory search, (3) each neuron cluster represents a swarm, which refines the promising sub-space regions identified by the symbiotic EA.

## V. DISCUSSION AND FUTURE WORK

The ANOVA analyses (table I) prove that the synergy between evolution (symbiotic EA) and learning (PSO) in SMNE significantly enhances learning efficiency. This synergy relies on an effective balance between exploratory (genetic search) and exploitative (lifetime learning) search in the solution space. Implicit fitness sharing preserves this synergetic balance by maintaining genetic diversity through induced niching pressure. Genetic diversity allows a continued exploratory search, without which lifetime learning could not exploit the solution space effectively. Niching pressure bounds the solution space into distinct sub-space regions (clusters in figure IV), which are partial solutions to the complete task. These sub-space regions are prime candidates for a local search (learning). The symbiotic EA's niching phenomenon thus aids learning by creating good conditions for lifetime learning (i.e. initial weights). Learning guides evolution by absolving it from exploring neighbouring solutions to the current evolutionary solutions. Evolution only needs to find appropriate regions in the solution space, rather than specific points in the solution space. A synergetic effect thus motivates the learning efficiency in the SMNE algorithm.

In the SMNE algorithm, evolution is learning at the level of the neuron population, while lifetime learning (PSO) is learning at the level of each individual neuron. The evolutionary task searches for cooperative neurons, while the learning task seeks to improve each neuron's partial solution to the complete task. The evolutionary and learning tasks are thus quite different. What the neurons are required to learn (learning task) and which neurons are selected during evolution (evolutionary task), are indirectly related. The evolutionary fitness surface and the learning fitness surface are correlated, i.e. superior neurons tend to have high fitness values on both the fitness surfaces [8]. A superior neuron cooperates effectively and also represents a good partial solution to the control task. Effective synergy results once high correlations between the learning and the evolutionary fitness surfaces are found.

However, the fitness landscapes of the learning and evolutionary tasks are continuously changing, relative to one another, during evolution. This continuous change depends on the population's current location in the solution space. This suggests a dynamic correlation between the two fitness surfaces [8]. Consider a novel neuron, with high learning task (partial solution) fitness. High partial solution fitness improves the likelihood of selection for genetic reproduction. Over several generations, the neuron is thus likely to obtain additional pointers from the network population. A greater number of neuron pointers translates into a higher cooperation (evolutionary) fitness. For effective synergy, a search for high dynamic correlation between the fitness surfaces must be maintained.

Future work will focus on further variations of the SMNE algorithm, including sub-population migration and culling in the particle swarms. In addition, the method will be tested in other real-world tasks that require both an efficient exploration and an accurate fine-tuning of the solutions.

## VI. CONCLUSION

The Symbiotic Memetic Neuro-Evolution (SMNE) algorithm is effective at developing neurocontrollers for use in highly non-linear process environments. Implicit fitness sharing maintains genetic diversity. Implicit fitness sharing's niching pressure accelerates the evolutionary search for solution sub-spaces that may be exploited by local search. Particle swarm optimisation effectively absolves the EA from devoting its resources to local refinements. The synergy between the symbiotic EA and PSO accelerates learning from dynamic process models. SMNE's efficient learning translates to greater economic return for the process industries.

## ACKNOWLEDGEMENTS

This work was supported in part by the South African Foundation for Research and Development, the Harry-Crossley Scholarship Fund, the National Science Foundation under grant IIS-0083776 and by the Texas Higher Education Coordinating Board under grant ARP-003658-476-2001..

## REFERENCES

[1] A.v.E. Conradie,"Neurocontroller development for nonlinear processes utilising evolutionary reinforcement learning", M.Sc. thesis, University of Stellenbosch, South Africa, 2000.

[2] D.D. Brengel and W.D. Seider, "Cooridnated design and control optimization of nonlinear processes", Computers and Chemical Engineering, 16(9), pp. 861-886, 1992.

[3] L.P. Kaelbling, M.L. Littman and A.W. Moore, "Reinforcement Learning: A Survey", Journal of Artificial Intelligence Research, 4, pp. 237-285, 1996.

[4] D.E. Moriarty and R. Miikkulainen, "Forming Neural Networks through Efficient and Adaptive Coevolution", Evolutionary Computation, 5(4), pp. 373-399, 1998.

[5] P. Merz, "Memetic algorithms for combinatorial optimization problems", Ph.D. thesis, University of Siegen, Germany, 2000.

[6] J. Horn, D.E. Goldberg and K. Deb, "Implicit niching in the learning classifier system: Nature's way", Evolutionary Computation, 2(1), pp. 37-66, 1994.

[7] Y. Shi and R.C. Eberhart, "Empirical study of particle swarm optimization", Proceedings of the 1999 Congress on Evolutionary Computation, IEEE Service Center, Piscataway, NJ, pp. 1945-1950, 1999.

[8] S. Nolfi, J.L. Elman and D. Parisi, "Learning and evolution in neural networks", Adaptive Behavior, (3), 1: 5-28, 1994.