# Simulating the Strategic Adaptation of Organizations using OrgSwarm

Anthony Brabazon[1], Arlindo Silva[2,3], Tiago Ferra de Sousa[3], Michael O'Neill[4], and Ernesto Costa[2]

[1] Faculty of Commerce, University College Dublin, Ireland.
anthony.Brabazon@ucd.ie
[2] Centro de Informatica e Sistemas da Universidade de Coimbra, Portugal.
ernesto@dei.uc.pt
[3] Escola Superior de Tecnologia, Instituto Politecnico de Castelo Branco, Portugal
arlindo@est.ipcb.pt
[4] Dept. Of Computer Science & Information Systems, University of Limerick, Ireland.
Michael.ONeill@ul.ie

**Abstract.** This chapter extends the particle swarm metaphor into the domain of organization science. A simulation model (*OrgSwarm*) is introduced which can be used to simulate the adaptation of a population of organizations on a strategic landscape. The simulator embeds a number of features including organizational inertia and dynamic landscapes. These features allow the examination of a wide range of real-life scenarios. The chapter also reports the results of a number of simulation experiments.

## 1 Introduction

In an organizational setting, a strategy consists of a choice of what activities the organization will perform, and choices as to how these activities will be performed [23]. These choices define the strategic configuration of the organization. Recent work by [20] and [25] has recognized that strategic configurations consist of interlinked individual elements (decisions), and have applied general models of interconnected systems such as Kauffman's NK model to examine the implications of this for processes of organizational adaptation.

Following a long-established metaphor of adaptation as search [29], strategic adaptation is considered in this study as an attempt to uncover peaks on a high-dimensional strategic landscape. Some strategic configurations produce high profits, others produce poor results. The search for good strategic configurations is difficult due to the vast number of strategic configurations possible, uncertainty as to the nature of topology of the strategic landscape faced by an organization, and changes in the topology of this landscape over time. Despite these uncertainties, the search process for good strategies is not blind. Decision-makers receive feedback on the success of their current and historic strategies, and can assess the payoffs received by the strategies of their competitors [18]. Hence, certain areas of the strategic landscape are illuminated.

Organizations do not exist in isolation but interact with, and receive feedback from their environment. Their efforts at strategic adaption are guided by *social* as well as *individual* learning. Good ideas discovered by one organization disseminate over time. Particle swarm algorithms also emphasize the importance of individual and social learning processes. Surprisingly, despite the parallels between the learning processes in particle swarm algorithms and those in populations of organizations, as yet the particle swarm metaphor has not been applied to the domain of organizational science. This chapter describes a novel simulation model based on the particle swarm metaphor, and applies this to examine the process of organizational adaptation. This study therefore extends the particle swarm metaphor into the domain of organization science.

## 2 Strategic Adaptation

Strategic adaptation and strategic inertia are closely linked. If strategic adaptation is problematic, inertia is a likely contributing cause. Broadly speaking, the strategic inertia of organizations stems from two sources, *imprinting forces*, and as a *consequence of market selection forces.*

Imprinting forces [3] combine to define and solidify the strategic configuration of a newly formed organization. These forces include the dominant initial strategy pursued by the organization, the skills / prior experience of the management team, and the distribution of decision-making influence in the organization at time of founding [3]. All of these influence the initial choice of organizational strategy. As consensus concerning the strategy emerges, it is imprinted on the organization through resource allocation decisions [26]. The imprinting leads to inertia by creating sunk costs, internal political constraints, and a rigid organizational structure. Over time this inertia intensifies due to the formation of an organizational history which creates barriers to industry exit, and legitimacy issues if adaptation is suggested [5]. The resulting inertia serves to circumscribe the organization's ability to adapt its strategy in the future. Imprinting also occurs as relationships are built up with suppliers and customers. The creation of a web of these relationships can serve to constrain the range of strategic alternatives in the future, as strategic moves which dramatically disrupt the web are less likely to be considered.

The discussion of strategic inertia was extended by [6] who posited that inertia is also created as a natural *consequence* of the market-selection process, claiming that 'selection processes tend to favor organizations whose structures are difficult to change.' (p. 149). The basis of this claim is that organizations which can produce a good or service reliably (consistently of a minimum quality standard) are favored for trading purposes by other organizations, and therefore by market selection processes. The routines required to produce a product or service reliably, tend to lead to structural inertia, as the construction of standarized routines leads to an increase in the complexity of the patterns of links between organizational subunits [6, 19]. It can therefore be posited that efficient organizations are likely to exhibit inertia. As organizations seek better environment-structure congruence, their systems become increasingly specialized and interlinked, making changes to their activities become costly and difficult. Tushman and O'Reilly [28] note that structural inertia is rooted in the size, complexity and interdependence of the firm's structures, systems, procedures and processes. Theoretical support for these assertions, that increasing organizational complexity can make adaptation difficult, is found in [10] and [25], as the heightened degree of interconnections between activities within the organization will increase the 'ruggedness' of the strategic landscape on which they are adapting.

## 3 Particle Swarm Algorithm

This section provides an introduction to the canonical Particle Swarm algorithm (PSA). The term PSA is used in place of the commonly-used PSO (Particle Swarm Optimization) in this chapter, as the object is not to develop a tool for 'optimizing', but to adapt and apply the swarm metaphor as a model of organizational adaptation. The PSA [12, 17] has been widely used for function optimization, and is based on a metaphor of human social interaction [14].

Under the swarm metaphor, a swarm of particles (entities) are assumed to move (fly) through an n-dimensional space, typically looking for a function optimum. Each particle is assumed to have two associated properties, a current position and a velocity. Each particle also has a memory of the best location in the search space that it has found so far (**pbest**), and knows the best location found to date by all the particles in the population (**gbest**). At each step of the algorithm, particles are displaced from their current position by applying a velocity vector to them. The size and direction of this velocity is influenced by the velocity in the previous iteration of the algorithm (simulates 'momentum'), and the current location of a particle relative to its **pbest** and **gbest**. Therefore, at each step, the size and direction of each particle's move is a function of its own history (experience), and the social influence of its peer group. A number of variants of the PSA exist.

**Description of PSA** The following paragraphs provide a description of the continuous version of the PSA. The algorithm is initially described narratively. This is followed by a description of the particle position-update equations.

i. Initialize each particle in the population by randomly selecting values for its location and velocity vectors
ii. Calculate the fitness value of each particle. If the current fitness value for a particle is greater than the best fitness value found for the particle so far, then revise **pbest**
iii. Determine the location of the particle with the highest fitness and revise **gbest** if necessary
iv. For each particle, calculate its velocity according to equation (1)
v. Update the location of each particle

vi. Repeat steps ii - v until stopping criteria are met

Each particle $i$ has an associated current position in $d$-dimensional space $\mathbf{x_i}$, a current velocity $\mathbf{v_i}$, and a personal best position $\mathbf{y_i}$. During each iteration of the algorithm, the location and velocity of each particle is updated using equations (1) - (4). Assuming a function $f$ is to be maximized, that the swarm consists of $n$ particles, and that $r_1$, $r_2$ are drawn from a uniform distribution in the range (0,1), the velocity update is described as follows

$$\mathbf{v_i}(t+1)=W\mathbf{v_i}(t)+c_1r_1(\mathbf{y_i}-\mathbf{x_i}(t))+c_2r_2(\mathbf{\hat{y}}-\mathbf{x_i}(t)) \tag{1}$$

where $\mathbf{\hat{y}}$ is the location of the global-best solution found by all the particles. A variant on the basic algorithm is to use a local rather than a global version of **gbest**, and the term **gbest** is replaced by **lbest**. In the local version, **lbest** is set independently for each particle, based on the best point found thus far within a *neighborhood* of that particle's current location.

In every iteration of the algorithm, each particle's velocity is stochastically accelerated towards its previous best position and towards **gbest** (or **lbest**). The weight-coefficients $c_1$ and $c_2$ control the relative impact of **pbest** and **gbest** locations on the velocity of a particle. The parameters $r_1$ and $r_2$ ensure that the algorithm is stochastic. A practical effect of the random coefficients $r_1$ and $r_2$, is that neither the individual nor the social learning terms are always dominant. Sometimes one or the other will dominate [17].

Although the velocity update has a stochastic component, the search process is not random. It is guided by the memory of past 'good' solutions (corresponding to a psychological tendency for individuals to repeat strategies which have worked for them in the past [14], and by the global best solution found by all particles thus far. $W$ represents a momentum coefficient which controls the impact of a particle's prior-period velocity on its current-period velocity. Each component (dimension) of the velocity vector $\mathbf{v_i}$ is restricted to a range $[-v_{max}, v_{max}]$ to ensure that individual particles do not leave the search space. The implementation of a $v_{max}$ parameter can also be interpreted as simulating the incremental nature of most learning processes [14]. The value of $v_{max}$ is usually chosen to be $k * x_{max}$, where $0 < k < 1$. Once the velocity update for particle $i$ is determined, its position is updated and **pbest** is updated if necessary, as described in equations 2-4.

$$\mathbf{x_i}(t+1)=\mathbf{x_i}(t)+\mathbf{v_i}(t+1) \tag{2}$$

$$\mathbf{y_i}(t+1)=\mathbf{y_i}(\mathbf{t}) \text{ if, } f(\mathbf{x_i}(t))\leq f(\mathbf{y_i}(t)), \tag{3}$$

$$\mathbf{y_i}(t+1)=\mathbf{x_i}(t) \text{ if, } f(\mathbf{x_i}(t))>f(\mathbf{y_i}(t)) \tag{4}$$

After all particles have been updated, a check is made to determine whether **gbest** needs to be updated.

$$\mathbf{\hat{y}}\in(\mathbf{y_0},\mathbf{y_1},\ldots,\mathbf{y_n})|f(\mathbf{\hat{y}})=\max\ (f(\mathbf{y_0}),f(\mathbf{y_1}),\ldots,f(\mathbf{y_n})) \tag{5}$$

**Particle Swarm as a Metaphor for Organizational Adaptation** Although particle swarm algorithms have been used extensively in function optimization (Particle Swarm Optimization), the original inspiration for PSAs arose from observations of animal and human social behavior [12]. Kennedy has published a series of papers which emphasize the social aspects of particle swarm [14–16] and this work was given prominence in the first major book on particle swarm [17].

The velocity update formula (equation 1) can be divided into *cognitive* and *social* components [14], with the former relating to the adaptive history of a particle, individual, or an organization. The cognitive term can be considered as an interpretation of Thorndike's *Law of Effect* [27], which states that a behavior which is followed by a (positive) reinforcement becomes more likely in the future, or in other words, learning from experience. The individual learning component in the velocity update formula $(\mathbf{y_i(t)} - \mathbf{x_i(t)})$ introduces a stochastic tendency to return to previously rewarded strategies, mimicking a psychological tendency for managers to repeat strategies which have worked for them in the past [14]. The social learning component of the formula $(\mathbf{\hat{y}_i(t)} - \mathbf{x_i(t)})$ bears comparison with social *no-trial learning* [1], where the observation of a peer being reinforced for a behavior, will increase the probability of the observer engaging in the same behavior.

The mechanisms of the canonical PSA bear *prima facie* similarities to those of the domain of interest, organizational adaptation. It adopts a populational perspective, and learning in the algorithm just as in populations of organizations, is both distributed and parallel. Organizations persist in employing already discovered good strategies, and are attracted to, and frequently imitate, good product ideas and business practices discovered by other organizations. However, the canonical PSA requires modification before it can employed as a component of a plausible simulation model of organizational adaptation. These modifications are discussed in the next section.

## 4 Simulation Model

The two key components of the simulation model, the landscape generator (environment), and the adaption of the basic Particle Swarm algorithm to incorporate the activities and interactions of the agents (organizations) are described next.

### 4.1 Strategic Landscape

In this study, the strategic landscape is defined using the NK model [9, 10]. It is noted *ab initio* that application of the NK model to define a strategic landscape is not atypical and has support from prior literature in organizational science which has adopted this approach [20, 25, 7, 24], and related work on technological innovation [21, 11]. The NK model considers the behavior of systems which are comprised of a configuration (string) of N individual elements. Each of these elements are in turn interconnected to K other of the N elements (K<N). In a

general description of such systems, each of the N elements can assume a finite number of states. If the number of states for each element is constant ($S$), the space of all possible configurations has N dimensions, and contains a total of $\prod_{i=1}^{N} S_i$ possible configurations.

In Kauffman's operationalization of this general framework [10], the number of states for each element is restricted to two (0 or 1). Therefore the configuration of N elements can be represented as a binary string . The parameter K, determines the degree of fitness interconnectedness of each of the N elements and can vary in value from 0 to N-1. In one limiting case where K=0, the contribution of each of the N elements to the overall fitness value (or worth) of the configuration are independent of each other. As K increases, this mapping becomes more complex, until at the upper limit when K=N-1, the fitness contribution of any of the N elements depends both on its own state, and the simultaneous states of all the other N-1 elements, describing a fully-connected graph.

If we let $s_i$ represent the state of an individual element $i$, the contribution of this element ($f_i$) to the overall fitness ($F$) of the entire configuration is given by $f_i(s_i)$ when K=0. When K>0, the contribution of an individual element to overall fitness, depends both on its state, and the states of K other elements to which it is linked ($f_i(s_i : s_{i1}, ..., s_{ik})$). A random fitness function (U(0,1)) is adopted, and the overall fitness of each configuration is calculated as the average of the fitness values of each of its individual elements.

Altering the value of K effects the ruggedness of the described landscape, and consequently impacts on the difficulty of search on this landscape [9, 10]. The strength of the NK model in the context of this study is that by tuning the value of K it can be used to generate strategic landscapes (graphs) of differing degrees of local-fitness correlation (ruggedness).

The strategy of an organization is characterized as consisting of N attributes [20]. Each of these attributes represents a strategic decision or policy choice, that an organization faces. Hence a specific strategic configuration **s**, is represented as a vector $s_1, ..., s_N$ where each attribute can assume a value of 0 or 1 [25]. The vector of attributes represents an entire organizational form, hence it embeds a choice of markets, products, method of competing in a chosen market, and method of internally structuring the organization [25]. Good consistent sets of strategic decisions - configurations, correspond to peaks on the strategic landscape.

The definition of an organization as a vector of strategic attributes finds resonance in the work of Porter [22, 23], where organizations are conceptualized as a series of activities forming a value-chain. The choice of what activities to perform, and subsequent decisions as to how to perform these activities, defines the strategy of the organization. The individual attributes of an organization's strategy interact. For example, the value of an efficient manufacturing process is enhanced when combined with a high-quality sales force. Differing values for K correspond to varying degrees of payoff-interaction among elements of the organization's strategy [25]. As K increases, the difficulty of the task facing strategic decision makers is magnified. Local-search attempts to improve an organization's position on the strategic landscape become ensnared in a web of conflicting constraints.

### 4.2 Simulation Model

Five characteristics of the problem domain which impact on the design of a simulation model are:

i. The environment is dynamic
ii. Organizations are prone to strategic inertia. Their adaptive efforts are anchored by their past
iii. Organizations do not knowingly select poorer strategies than the one they already have (election operator)
iv. Organizations make errorful *ex-ante* and assessments of fitness
v. Organizations co-evolve

Although our simulator embeds all of the above, in this chapter we report results which consider the first three of these factors. We note that this model bears passing resemblance to the *eleMentals* model of [15], which combined a swarm algorithm and an NK landscape, to investigate the development of culture and intelligence in a population of hypothetical beings called eleMentals. However, the *OrgSwarm* simulator is differentiated from the eleMental model on grounds of application domain, and because of the incorporation of the above characteristics of the domain.

**Dynamic environment** Organizations do not compete in a static environment. The environment may alter as a result of exogenous events, for example a *regime change* such as the emergence of a new technology, or a change in customer preferences. This can be mimicked in the simulation by stochastically respecifing the strategic landscape during the course of a simulation run. These respecifications simulate a dynamic environment, and a change in the environment may at least partially negate the value of past learning (adaptation) by organizations. Minor respecifications are simulated by altering the fitness values associated with one of the N dimensions in the NK model, whereas in major changes, the fitness of the entire NK landscape is redefined. The environment faced by organizations can also change as a result of competition between the population of organizations. The effect of inter-firm competition is left for future work.

**Strategic Anchor** Organizations do not have complete freedom to alter their current strategy. Their adaptive processes are subject to strategic inertia. This inertia springs from the organization's culture, history, and the mental models of its management [3]. In the simulation, strategic inertia is mimicked by implementing a *strategic anchor*. The degree of inertia can be varied from zero to high. In the latter case, the organization is highly constrained from altering its strategic stance. By allowing the weight of this anchor to vary, adaptation processes corresponding to different industries, each with different levels of inertia, can be simulated. Inertia could be incorporated into the PSA in a variety of ways. We have chosen to incorporate it into the velocity update equation, so that the velocity and direction of the particle at each iteration is also a function of the location of its strategic anchor. Therefore for the simulations, equation 1 is altered by adding an additional 'anchor' term

$$\mathbf{v_i}(t+1)=\mathbf{v_i}(t)+R_1(\mathbf{y_i}-\mathbf{x_i}(t))+R_2(\hat{\mathbf{y}}-\mathbf{x_i}(t)+R_3(\mathbf{a_i}-\mathbf{x_i}(t)) \tag{6}$$

where $\mathbf{a_i}$ represents the position of the anchor for organization $i$ (a full description of the other terms such as $R_1$ is provided in the pseudo-code below). The weight attached to the anchor parameter ($R_3$) (relative to those attached to pbest and gbest), can be altered by the modeler. The position of the anchor can be fixed at the initial position of the particle at the start of the simulation, or it can be allowed to 'drag', thereby being responsive to the adaptive history of the particle. In the latter case, the position of the anchor for each particle corresponds to the position of that particle '$x$' iterations ago.

**Election operator** Real-world organizations do not usually intentionally move to poorer (lower payoff) strategies than the one they already have. Hence, an election operator (also referred to as a *conditional update* or *ratchet operator*) is implemented, which when turned on ensures that position updates which would worsen an organization's strategic fitness are discarded. In these cases, an organization remains at its current location.

## 4.3 Outline of Swarm Algorithm

As the strategic landscape is described using a binary representation (the NK model), the canonical PSA is adapted for the binary case using the *BinPSO* version of the algorithm [13]. The binary version of the PSA is inspired by the idea that an agent's probability of making a binary decision (yes/no, true/false) is a function of both personal history and social factors. The probability that an agent chooses a value of (for example) 1 for a particular decision in the next time period, is a function of the agent's history ($\mathbf{x_i(t)}, \mathbf{v_i(t)}$ & $\mathbf{pbest}$), and social factors ($\mathbf{lbest}$) (see equation 7).

$$Prob(\mathbf{x_i(t+1)}=1)=f(\mathbf{x_i(t)},\mathbf{v_i(t)},\mathbf{pbest},\mathbf{lbest}) \tag{7}$$

The vector $\mathbf{v_i}$ is interpreted as organization $i$'s predisposition to set each of the $N$ binary strategic choices that it faces to one. The higher the value of $v_i^j$ for an individual decision $j$, the more likely that organization $i$ will choose to set decision $j=1$, with lower values of $v_i^j$ favoring the choice of decision $j=0$.

In order to model the tendency of managers to repeat historically good strategies, values for each dimension of $\mathbf{x_i}$ which match those of $\mathbf{pbest}$, should become more probable in the future. Adding the difference between $pbest_i^j$ and $x_i^j$ for organization $i$ to $v_i^j$ will increase the likelihood that organization $i$ will choose to set decision $j=1$ if the difference is positive (when $pbest_i^j=1$ and $x_i^j=0$). If the difference between $pbest_i^j$ and $x_i^j$ for organization $i$ is negative (when $pbest_i^j = 0$, and $x_i^j = 1$), adding the difference to $v_i^j$ will decrease $v_i^j$.[5]

In each iteration of the algorithm, the agent adjusts his decision-vector ($\mathbf{x_i(t)}$), taking account of his historical experience (**pbest**), and the best strategy found by his peer-group (**lbest**). Hence, the velocity update equation used in the continuous version of the PSA (see equation 6) can still be used, although now, $\mathbf{v_i(t+1)}$ is interpreted as the updated vector of an agent's predisposition (or probability thresholds) to set each of the $N$ binary strategic choices that it faces to one.

$$\mathbf{v_i(t+1)} = \mathbf{v_i(t)} + R_1(\mathbf{pbest_i} - \mathbf{x_i(t)}) + R_2(\mathbf{lbest_i} - \mathbf{x_i(t)} + R_3(\mathbf{anchor_i} - \mathbf{x_i(t)}) \tag{8}$$

To ensure that each element of the vector $\mathbf{v_i(t+1)}$ is mapped into (0,1), a sigmoid transformation is performed on each element $j$ of $\mathbf{v_i(t+1)}$ (see equation 9).

$$Sig(v_i^j(t+1)) = \frac{1}{1 + exp(-v_i^j(t+1))} \tag{9}$$

Finally, the transformed vector of probability thresholds is used to determine the values of each element of $x_i(t+1)$, by comparing each element of $Sig(\mathbf{v_i(t)})$ with a random number drawn from $U(0,1)$ (see equation 10).

$$\text{If } U(0,1) < Sig(v_i^j(t+1)), \text{ then } x_i^j(t+1) = 1; \text{ else } x_i^j(t+1) = 0 \tag{10}$$

In the binary version of the algorithm, trajectories / velocities are changes in the probability that a coordinate will take on a zero or a one value. $Sig(v_i^j)$ represents the probability of bit $x_i^j$ taking the value 1 [13]. Therefore, if $Sig(v_i^j) = 0.3$ there is a thirty percent chance that $x_i^j = 1$, and a seventy percent chance it is zero.

**Pseudocode for Algorithm** The pseudo-code for the swarm algorithm in the simulator is as follows:

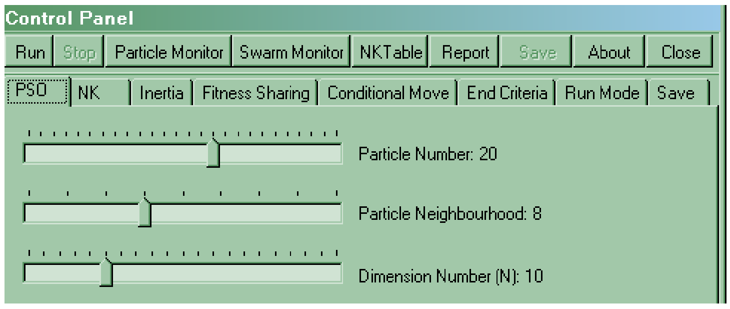
[5] The difference in each case is weighted by a random number drawn from U(0,1). Therefore, if $pbest_i^j = 1$, $(pbest_i^j - x_i^j) * U(0,1)$ will be non-negative. Adding this to $v_i^j$ will increase $v_i^j$, and therefore also increase the probability that $x_i^j = 1$. On the other hand if $pbest_i^j = 0$, $v_i^j$ will tend to decrease, and $Prob(x_i^j) = 1$ becomes smaller.

```
For each entity in turn

For each dimension (strategic decision) n

v[n]=v[n]+R1*(pbest[n]-x[n])+R2*(lbest[n]-x[n])+R3*(a[n]-x[n])
If(v[n]>Max) v[n]=Vmax
    If(v[n]<-Vmax) v[n]=-Vmax
    If(Pr<Sig(v[n]))t[n]=1
    Else t[n]=0
If(fitness(t)*(1+e))>fitness(x))  //ratchet operator
    For each dimension n
        x[n]=t[n]
UpdateAnchor(a)        //if iteratively update anchor
                       //option is selected
```

$R_1$, $R_2$ and $R_3$ are random weights drawn from a uniform distribution ranging from 0 to $R_{1max}$, $R_{2max}$ and $R_{3max}$ respectively, and they weight the importance attached to pbest, lbest and anchor in each iteration of the algorithm. $R_{1max}$, $R_{2max}$ and $R_{3max}$ are constrained to sum up to 4.0 in line with the *BinPSO* alogrithm of [13]. $x$ is the particle's actual position, *pbest* is its past best position, *lbest* its local best and $a$ is the position of its anchor. $V_{max}$ is set to 4.0 to ensure that $Sig(v[n])$ does not get too close to either 0 or 1, therefore ensuring that there is a non-zero possibility that a bit will flip state during each iteration. $Pr$ is a random value drawn from U(0,1), and $Sig$ is the sigmoid function: $Sig(x) = \frac{1}{1+exp(-x)}$, which squashes $v$ into the range $0 \rightarrow 1$ range. $t$ is a temporary record which is used in order to implement the ratchet operator. If the new strategy is considered better than the organization's existing strategy, it is accepted and $t$ is copied into $x$. Otherwise $t$ is discarded and $x$ remains unchanged. $e$ is the error or noise, injected in the fitness evaluation, in order to mimic an errorful forecast of strategy fitness.



Particle Number: 20

Particle Neighbourhood: 8

Dimension Number (N): 10

**Fig. 1.** Main control screen for OrgSwarm.

## 4.4 Simulator

Although the underlying code for the *OrgSwarm* simulator is written in C++, the user interacts with the simulator through a series of easy-to-use screens (Fig. 1 shows one of the screens in the main control menu for the simulator). These screens allow the user to select and alter a wide variety of parameters which determine the nature of the simulation run. In essence, the simulator allows the user to select choices for four items:

i. the form of NK landscape generated,
ii. the nature of the search heuristics to be employed by inventors,
iii. the number of simulations to be run, and
iv. the form of output generated during the simulation run.

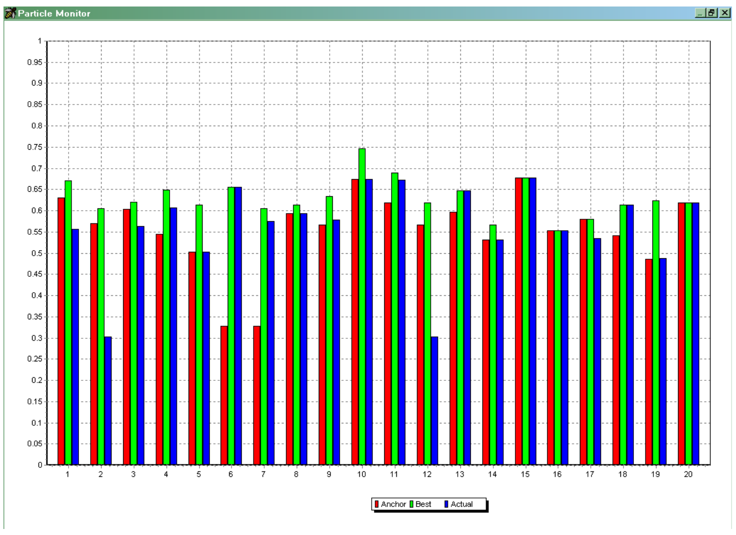
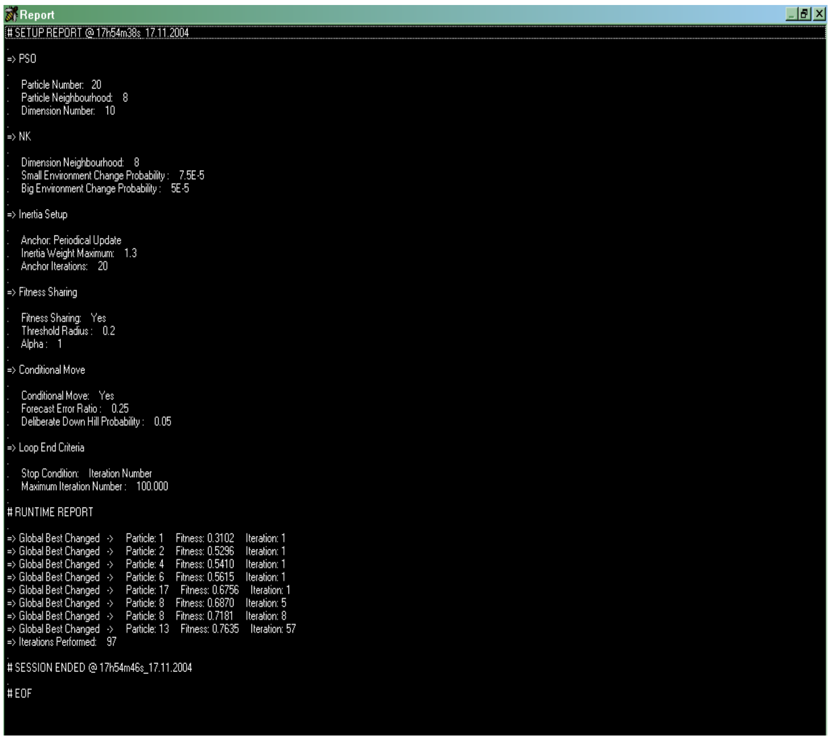


**Fig. 2.** OrgSwarm screendump showing the status of each particle in the population during the simulation run. Three bars are shown for each of the twenty particles in the population, and these bars represent the fitness of the anchor location, the fitness of the pbest location, and the fitness of the current location of each particle.

During the simulation run, a series of graphics (see Fig. 2 for an example of a graphic which shows the status of each particle in the population during the simulation run), and a run report (see Fig. 3) can be displayed. The report display records the full list of simulation parameters chosen by the modeller, as well as providing a running record of the best design in the population at the end of each iteration. The simulator also facilitates the recording of comprehensive run-data to disk during the simulation.



```
Report
# SETUP REPORT @ 17h54m38s_17.11.2004
.
=> PSO
.
.    Particle Number:   20
.    Particle Neighbourhood:    8
.    Dimension Number:    10
.
=> NK
.
.    Dimension Neighbourhood:    8
.    Small Environment Change Probability :    7.5E-5
.    Big Environment Change Probability :    5E-5
.
=> Inertia Setup
.
.    Anchor: Periodical Update
.    Inertia Weight Maximum:    1.3
.    Anchor Iterations:    20
.
=> Fitness Sharing
.
.    Fitness Sharing:    Yes
.    Threshold Radius :    0.2
.    Alpha :    1
.
=> Conditional Move
.
.    Conditional Move:    Yes
.    Forecast Error Ratio :    0.25
.    Deliberate Down Hill Probability :    0.05
.
=> Loop End Criteria
.
.    Stop Condition:    Iteration Number
.    Maximum Iteration Number :    100.000
.
# RUNTIME REPORT
.
=> Global Best Changed  ->    Particle: 1    Fitness: 0.3102    Iteration: 1
=> Global Best Changed  ->    Particle: 2    Fitness: 0.5296    Iteration: 1
=> Global Best Changed  ->    Particle: 4    Fitness: 0.5410    Iteration: 1
=> Global Best Changed  ->    Particle: 6    Fitness: 0.5615    Iteration: 1
=> Global Best Changed  ->    Particle: 17    Fitness: 0.6756    Iteration: 1
=> Global Best Changed  ->    Particle: 8    Fitness: 0.6870    Iteration: 5
=> Global Best Changed  ->    Particle: 8    Fitness: 0.7181    Iteration: 8
=> Global Best Changed  ->    Particle: 13    Fitness: 0.7635    Iteration: 57
=> Iterations Performed:    97
.
# SESSION ENDED @ 17h54m46s_17.11.2004
.
# EOF
```

**Fig. 3.** A typical run report generated by OrgSwarm.

## 5 Results

All simulations were run for 5,000 iterations, and all reported fitnesses are the average population fitnesses, and average environment best fitnesses, across 30 separate simulation runs. On each of the simulation runs, the NK landscape is specified anew, and the positions and velocities of particles are randomly initialized at the start of each run. A population of 20 particles is employed, with a neighborhood of size 18. The choice of a high value for the neighborhood, relative to the size of the population, arises from the observation that real-world organizations know the profitability of their competitors.

Tables 1 and 2 provide the results for each of fourteen distinct PSA variants, at the end of 5,000 iterations, across a number of static and dynamic NK landscape scenarios. In each scenario, the same series of simulations are undertaken. Initially, a basic PSA is employed, without an anchor or a ratchet (conditional move) heuristic. This simulates a population of organizations searching a strategic landscape, where the population has no strategic inertia, and where organizations do not utilize a ratchet operator in deciding whether to alter their position on the strategic landscape.

The basic PSA is then supplemented by inclusion of a series of strategic anchor formulations, ranging from an anchor which does not change position during the simulation (initial position anchor) to one which can adapt after a time-lag (moving anchor). Two lag periods are examined, 20 and 50 iterations. Differing weights can be attached to the anchor term in the velocity equation 6, ranging from 0 (anchor is 'turned off') to a maximum of 4. To determine whether the weight factor for the anchor term has a critical impact on the results, results are reported for weight values of both 1 and 3, corresponding to low and high inertia weights. Next, to isolate the effect of the ratchet, the conditional move operator is implemented, and the anchor term is dropped. Finally, to ascertain the combined effect of both ratchet and anchor, the anchor simulations outlined above are repeated with the ratchet operator 'turned on'.

'Real world' strategy vectors consist of a large array of strategic decisions. A value of N=96 was chosen in defining the landscapes in this simulation. It is noted that there is no unique value of N that could have been selected, but the selection of very large values are not feasible due to computational limitations. However, a binary string of 96 bits provides $2^{96}$, or approximately $10^{28}$, distinct choices of strategy. It is also noted that we would expect the dimensionality of the strategy vector to exceed the number of organizations in the population, hence the size of the population is kept below 96, and a value of 20 is chosen. A series of landscapes of differing K values (0,4 and 10), representing differing degrees of fitness inter-connectivity, were used in the simulations.

## 5.1 Static Landscape

Table 1 and figures 4 and 5 provide the results for a static NK landscape. Examining these results suggests that the basic PSA, without anchor or ratchet heuristics, performs poorly, even on a static landscape. The average populational fitnesses obtained after 5,000 iterations (averaged over all 30 runs) is no better than random search, suggesting that unfettered adaptive efforts, based on 'social communication' between organizations (gbest), and a memory of good past strategies (pbest) is not sufficient to achieve high levels of populational fitness. When various anchor term mechanisms, simulating strategic inertia, are added to the basic PSA, the results are not qualitatively altered from those of the basic PSA. This suggests that social communication and inertia, are not sufficient for the attainment of high levels of populational strategic fitness.

When a ratchet heuristic is added to the basic PSA, a significant improvement (statistically significant at the 5% level) in both average populational, and average environment best fitness is obtained across landscapes of all K values, suggesting that the simple decision heuristic of 'only abandon your current strategy for a better one' can lead to notable increases in populational fitness.

Finally, the results of a series of simulations which combine anchor and ratchet mechanisms are reported. Virtually all of these combinations lead to significantly (at the 5% level) enhanced levels of populational fitness against the ratchet-only PSA, *suggesting that strategic inertia can be beneficial, when organizations employ a conditional move test before adopting new strategies.* Examining the combined ratchet and anchor results in more detail, the best results are obtained when the anchor is not fixed at the initial location of each particle on the landscape, but when it is allowed to 'drag' or adapt, over time. The results are not qualitatively sensitive to the weight value (1 or 3).

### 5.2 Dynamic Landscape

The real world is rarely static, and changes in the environment can trigger adaptive behavior by agents in a system [2]. Table 2 and figures 6 and 7 provide results for the case where the entire NK landscape is respecified in any iteration with a prob=0.00025. When the landscape is wholly or partially respecified, the benefits of past strategic learning by organizations is eroded (see [4, 8, 2] for a detailed discussion of the utility of the PSO in tracking dynamic environments).

Qualitatively, the results in both scenarios are similar to those obtained on the static landscape. The basic PSA, even if supplemented by an anchor mechanism, does not perform any better than random search. Supplementing the basic PSA with the ratchet mechanism leads to a significant improvement in populational fitness, with a further improvement in fitness occurring when the ratchet is combined with an anchor mechanism. In the latter case, an adaptive or dragging anchor gives better results than a fixed anchor, but the results between differing forms of dragging anchor do not show a clear dominance for any particular form. As for the static landscape case, the results for the combined ratchet / anchor, are relatively insensitive to the choice of weight value (1 or 3).

| **Algorithm** | **Fitness** | | |
|---|---|---|---|
| | **(N=96, K=0)** | **(N=96, K=4)** | **(N=96, K=10)** |
| Basic PSA | 0.4641 (0.5457) | 0.5002 (0.6000) | 0.4991 (0.6143) |
| Initial Anchor, w=1 | 0.4699 (0.5484) | 0.4921 (0.5967) | 0.4956 (0.6102) |
| Initial Anchor, w=3 | 0.4943 (0.5591) | 0.4994 (0.5979) | 0.4991 (0.6103) |
| Mov. Anchor (50,1) | 0.4688 (0.5500) | 0.4960 (0.6003) | 0.4983 (0.6145) |
| Mov. Anchor (50,3) | 0.4750 (0.5631) | 0.4962 (0.6122) | 0.5003 (0.6215) |
| Mov. Anchor (20,1) | 0.4644 (0.5475) | 0.4986 (0.6018) | 0.5001 (0.6120) |
| Mov. Anchor (20,3) | 0.4677 (0.5492) | 0.4994 (0.6156) | 0.4994 (0.6229) |
| Ratchet PSA | 0.5756 (0.6021) | 0.6896 (0.7143) | 0.6789 (0.7035) |
| Rach-Initial Anchor, w=1 | 0.6067 (0.6416) | 0.6991 (0.7261) | 0.6884 (0.7167) |
| Rach-Initial Anchor, w=3 | 0.5993 (0.6361) | 0.6910 (0.7213) | 0.6844 (0.7099) |
| Rach-Mov. Anchor (50,1) | 0.6659 (0.6659) | 0.7213 (0.7456) | 0.6990 (0.7256) |
| Rach-Mov. Anchor (50,3) | 0.6586 (0.6601) | 0.7211 (0.7469) | 0.6992 (0.7270) |
| Rach-Mov. Anchor (20,1) | 0.6692 (0.6695) | 0.7211 (0.7441) | 0.6976 (0.7243) |
| Rach-Mov. Anchor (20,3) | 0.6612 (0.6627) | 0.7228 (0.7462) | 0.6984 (0.7251) |

**Table 1.** Average (environment best) fitness after 5,000 iterations, static landscape.

## 6 Conclusions

In this chapter, a synthesis of a strategic landscape defined using the NK model, and a Particle Swarm metaphor is used to create a novel simulation model of the

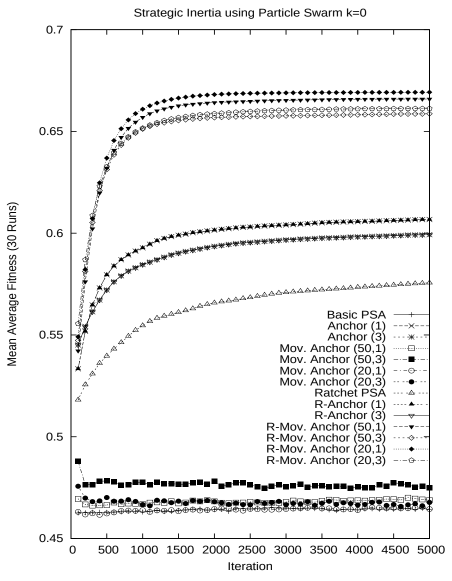
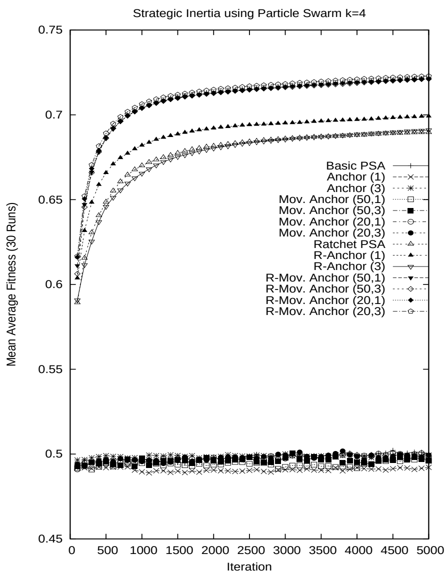


**Fig. 4.** Plot of the mean average fitness on the static landscape where k=0.



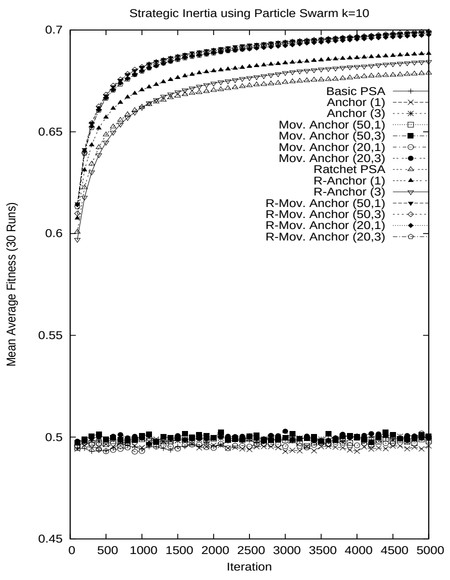


**Fig. 5.** Plot of the mean average fitness on the static landscape where k=4 (left) and 10 (right).

process of strategic adaptation of organizations. The results suggest that a degree of strategic inertia, in the presence of an election operator, can assist rather than

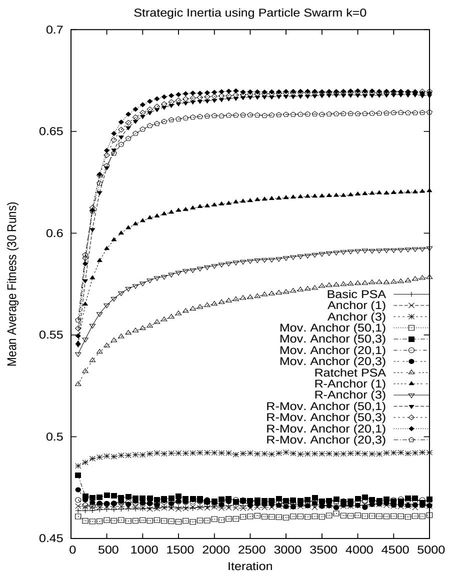


**Fig. 6.** Plot of the mean average fitness on the dynamic landscape where k=0.

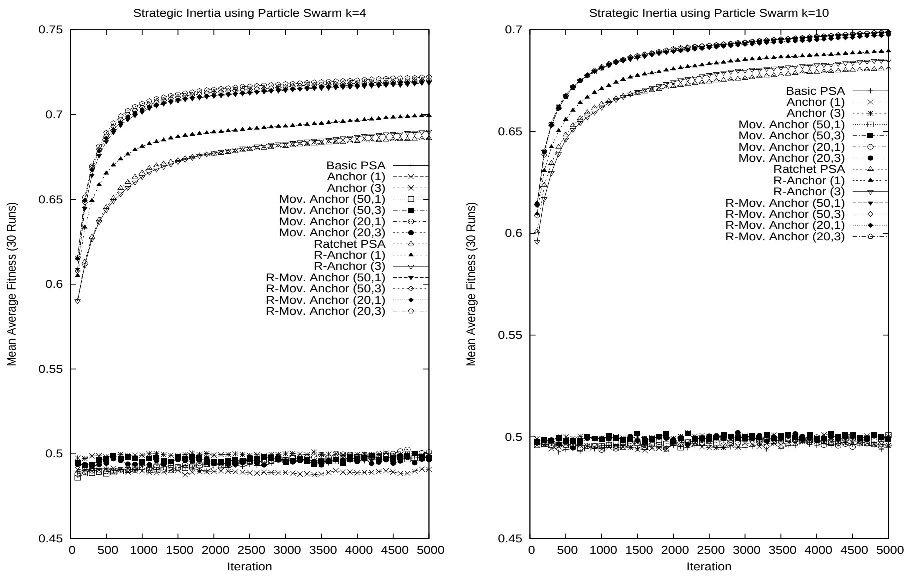


**Fig. 7.** Plot of the mean average fitness on the dynamic landscape where k=4 (left) and 10 (right).

hamper the adaptive efforts of populations of organizations in static and slowly changing strategic environments. The results also suggest that despite the claim

| **Algorithm** | **Fitness** | | |
|---|---|---|---|
| | **(N=96, K=0)** | **(N=96, K=4)** | **(N=96, K=10)** |
| Basic PSA | 0.4761 (0.5428) | 0.4886 (0.5891) | 0.4961 (0.6019) |
| Initial Anchor, w=1 | 0.4819 (0.5524) | 0.4883 (0.5822) | 0.4982 (0.6075) |
| Initial Anchor, w=3 | 0.5021 (0.5623) | 0.4967 (0.5931) | 0.4998 (0.6047) |
| Mov. Anchor (50,1) | 0.4705 (0.5450) | 0.4894 (0.5863) | 0.4974 (0.6008) |
| Mov. Anchor (50,3) | 0.4800 (0.5612) | 0.4966 (0.6053) | 0.5010 (0.6187) |
| Mov. Anchor (20,1) | 0.4757 (0.5520) | 0.4926 (0.5867) | 0.4985 (0.6097) |
| Mov. Anchor (20,3) | 0.4824 (0.5632) | 0.4986 (0.6041) | 0.5004 (0.6163) |
| Ratchet PSA | 0.5877 (0.6131) | 0.6802 (0.7092) | 0.6754 (0.7015) |
| Rach-Initial Anchor, w=1 | 0.6187 (0.6508) | 0.6874 (0.7180) | 0.6764 (0.7070) |
| Rach-Initial Anchor, w=3 | 0.6075 (0.6377) | 0.6841 (0.7130) | 0.6738 (0.7017) |
| Rach-Mov. Anchor (50,1) | 0.6517 (0.6561) | 0.7134 (0.7387) | 0.6840 (0.7141) |
| Rach-Mov. Anchor (50,3) | 0.6597 (0.6637) | 0.7049 (0.7304) | 0.6925 (0.7225) |
| Rach-Mov. Anchor (20,1) | 0.6575 (0.6593) | 0.7152 (0.7419) | 0.6819 (0.7094) |
| Rach-Mov. Anchor (20,3) | 0.6689 (0.6700) | 0.7158 (0.7429) | 0.6860 (0.7147) |

**Table 2.** Average (environment best) fitness after 5,000 iterations, entire landscape respecified stochastically.

for the importance of social learning in populations, social learning alone is not always enough, unless learnt lessons can be maintained by means of an election mechanism.

It is not possible in a single set of simulation experiments to exhaustively examine every possible combination of settings for each parameter in the simulation model. Future work will extend the range of settings examined. However, the initial results cast an interesting light on the role of anchoring in organizational adaptation, and the development of the swarm-landscape simulator extends the methodologies available to researchers to conceptualize and examine organizational adaptation.

Finally, it is noted that the concept of anchoring developed in this chapter is not limited to organizations, but is plausibly a general feature of social systems. Hence, the extension of the social swarm model to incorporate inertia may prove useful beyond this study.

## References

1. Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*, Englewood Cliffs, New Jersey: Prentice Hall.
2. Blackwell, T. (2003). Swarms in Dynamic Environments, *Proceedings of GECCO 2003*, Lecture Notes in Computer Science (2723), Springer-Verlag, Berlin, pp. 1-12.
3. Boeker, W. (1989). Strategic Change: The Effects of Founding and History, *Academy of Management Journal*, 32(3):489-515.
4. Eberhart, R. and Shi, Y. (2001). Tracking and optimizing dynamic systems with particle swarms, *in Proceedings of the CEC 2001*, 94-97:IEEE Press.
5. Hannan, M. and Freeman, J. (1977). The Populational Ecology of Organizations, *American Journal of Sociology*, 82(5): 929-964.
6. Hannan, M. and Freeman, J. (1984). Structural Inertia and Organizational Change, *American Sociological Review*, 49:149-164.
7. Gavetti, G. and Levinthal, D. (2000). Looking Forward and Looking Backward: Cognitive and Experiential Search, *Administrative Science Quarterly*, 45:113- 137.

8. Hu, X. and Eberhart, R. (2002). Adaptive particle swarm optimization: detection and response to dynamic systems, *in Proceedings of CEC 2002*, 1666-1670:IEEE Press.
9. Kauffman, S. and Levin, S. (1987). Towards a General Theory of Adaptive Walks on Rugged Landscapes, *Journal of Theoretical Biology*, 128:11-45.
10. Kauffman, S. (1993). *The Origins of Order*, Oxford,England: Oxford University Press.
11. Kauffman, S., Lobo, J. and MacReady, W. (1998). Optimal Search on a Technology Landscape, *Santa Fe Institute Working Paper 98-10-091.*
12. Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization, *Proceedings of the IEEE International Conference on Neural Networks*, December 1995, pp.1942-1948.
13. Kennedy, J. and Eberhart, R. (1997). A discrete binary version of the particle swarm algorithm, *Proceedings of the Conference on Systems, Man and Cybernetics*, pp. 4104-4109:IEEE Press.
14. Kennedy, J. (1997). The particle swarm: Social adaptation of knowledge, *Proceedings of the International Conference on Evolutionary Computation*, pp. 303-308:IEEE Press.
15. Kennedy, J. (1999). Minds and Cultures: Particle Swam Implications for Beings in Sociocognitive Space, *Adaptive Behavior*, 7(3/4):269-288.
16. Kennedy, J. (1999). Small worlds and mega-minds: effects of neighbourhood topology on particle swarm performance, *Proceedings of the International Conference on Evolutionary Computation*, 1931-1938, IEEE Press.
17. Kennedy, J., Eberhart, R. and Shi, Y. (2001). *Swarm Intelligence*, San Mateo, California: Morgan Kauffman.
18. Kitts, B., Edvinsson, L. and Beding, T. (2001). Intellectual capital: from intangible assets to fitness landscapes, *Expert Systems with Applications*, 20:35-50.
19. Levinthal, D. (1991). Random Walks and Organisational Mortality, *Administrative Science Quarterly*, 36:397-420.
20. Levinthal, D. (1997). Adaptation on Rugged Landscapes, *Management Science*, 43(7):934-950.
21. Lobo, J. and MacReady, W. (1999). Landscapes: A Natural Extension of Search Theory, *Santa Fe Institute Working Paper 99-05-037.*
22. Porter, M. (1985). *Competitive Advantage:Creating and Sustaining Superior Performance*, New York, The Free Press.
23. Porter, M. (1996). What is Strategy?, *Harvard Business Review*, Nov-Dec, 61-78.
24. Porter, M. and Siggelkow, N. (2001). Contextuality within Activity Systems,*Harvard Business School Working Paper Series*, No. 01-053, 2001.
25. Rivkin, J. (2000). Imitation of Complex Strategies, *Management Science*, 46(6):824- 844.
26. Stuart, T. and Podolny, J. (1996). Local Search and the Evolution of Technological Capabilities, *Strategic Management Journal*, 17:21-38.
27. Thorndike, E. (1911). *Animal Intelligence*, Macmillian, New York.
28. Tushman, M. and O'Reilly, C. (1996). Ambidextrous Organizations: Managing Evolutionary and Revolutionary Change, *California Management Review*, 38(4):8-30.
29. Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding and selection in evolution, *Proceedings of the Sixth International Congress on Genetics*, 1:356-366.