Chapter 16
Computational Motivation, Autonomy and Trustworthiness: Can We Have It All?

Kathryn Merrick, Adam Klyne and Medria Hardhienata

16.1 Autonomous Systems

In the past fifty years we have quickly moved from controlled systems to supervised systems [6], automatic systems and autonomous systems [23, 26]. Autonomous systems are highly adaptive systems that sense the environment and learn to make decisions about their own actions. They may display a high degree of proactivity, self-organization or self-motivation [31, 43], in reaching their objectives.

Autonomous systems may operate without the presence of a human. Alternatively, they may communicate, cooperate, and negotiate with humans to reach their goals. Thus, a complementary strand of research over the past decades has studied such man-computer symbiosis [25], including research that studies systems that can adapt their own level of automation [32], and systems that can achieve cognitive-cyber symbiosis [2].

There is a clear benefit for society if repetitive or dangerous tasks can be performed by machines. Yet, there are also barriers to the adoption of increasingly sophisticated technology. These barriers include both functionality related concerns—particularly in extreme, severe, complex and dynamic environments—as well as legal, ethical, social, safety and regulatory concerns [1].

In fact, many of these issues are related in some way to the level of trust held in autonomous system technologies. Trust is a pervasive concept that influences decision-making when the actions of one system (or agent[1]) can have an impact on another agent [3, 34]. Many definitions of trust have been proposed [4, 17, 22, 27]. At one level, trust can be defined as social contract between two agents [4]. A truster delegates a task to a trustee, and assumes the risk that the trustee might be untrustworthy. The trustee accepts the task, implicitly or explicitly promising to be trustworthy. The truster's decision to trust the trustee is influenced by the truster's

[1] Agent here can refer to a human, organization, or software system.

K. Merrick (✉) · A. Klyne · M. Hardhienata
School of Engineering and Information Technology,
University of New South Wales, Canberra, Australia
e-mail: k.merrick@adfa.edu.au

H. A. Abbass et al. (eds.), *Foundations of Trusted Autonomy*, Studies in Systems, Decision and Control 117, https://doi.org/10.1007/978-3-319-64816-3_16

attitude towards risk.[2] Trust involves the judgement of the truster in relation to the trustee agent based on the integration of the truster's cognitive attributes and life experience.

This chapter considers the impact of one of the emerging mechanisms for achieving autonomy—computational motivation—on the trustworthiness of autonomous systems. Motivation is the cause of action in natural systems (such as humans) [18]. Like trust, motivation has been defined from different perspectives. For motivation, this includes perspectives of drive, arousal, risk attitude, social attitude, expectancy, incentive, trait theory, attribution theory and approach-avoidance theory. Also like trust, motivation is understood to be influenced by the integration of an agent's cognitive attributes and life experience.

The concepts of motivation and trust overlap at least along the dimensions of risk attitude, social attitude and assimilation of life experience and cognitive attributes. (1) Agents with different motive profiles may act differently in the same situation as a result of different life experiences; (2) Differences in the motive profiles of agents (including risk and social attitude) may affect their ability to trust; and (3) Differences in the actions of agents with different motives may affect their trustworthiness.

This chapter will focus primarily on points (1) and (3) above. First, we consider the implications of motivation for functionality (Sect. 16.3) and then the implications for trustworthiness (Sect. 16.4). Point (2) above has not been widely examined from a computational perspective. To make our discussion concrete, in this chapter we consider these issues in the context of intrinsically motivated agent swarms. Many key variants of computational motivation have been considered for use in swarm systems, making this a timely and relevant for discussion. Section. 16.2 begins by providing an overview of the theory underlying the use of computational motivation in swarms of artificial agents, including a uniform notation for three intrinsically motivated swarm algorithms.

16.2 Intrinsically Motivated Swarms

At the heart of computational models of flocks, herds, schools, swarms and crowd behavior is Reynold's iconic boids model [35]. The boids model can be viewed as a kind of rule-based reasoning in which rules take into account certain properties of other agents. The three fundamental rules are:

- **Cohesion:** Each agent moves toward the average position of its neighbors;
- **Alignment:** Each agent steers so as to align itself with the average heading of its neighbors;
- **Separation:** Agents move to avoid hitting their neighbors.

[2]Risk here is the potential of losing something of value, weighed against the potential to gain something of value (an incentive) [8].

Each *boid* in a computational swarm applies these three rules at each time step. The rules are implemented as forces that act on agents when a certain condition holds. Suppose we have a group of n agents A^1, A^2, $A^3 \ldots A^n$. At time t each agent A^j has a position, x_t^j, and a velocity, v_t^j. x_t^j is a point and v_t^j is a vector. At each time step t, the velocity of each agent is updated as follows:

$$v_{(t+1)}^j = W_d v_t^j + W_c c_t^j + W_a a_t^j + W_s s_t^j \tag{16.1}$$

c_t^j is a vector in the direction of the average position of agents within a certain range of A^j (called the neighbours of A^j); a_t^j is a vector in the average direction of agents within a certain range of A^j; and s_t^j is a vector in the direction away from of the average position of agents within a certain range of A^j. These vectors are the result of cohesive, alignment and separation forces corresponding to the rules outlined above. Weights W_c,W_a and W_s strengthen or weaken the corresponding force. W_d strengthens or weakens the perceived importance of the *boid*'s existing velocity. Once a new velocity has been computed, the position of each agent is updated by:

$$x_{(t+1)}^j = x_t^j + v_{(t+1)}^j \tag{16.2}$$

As noted above, agents that are within a certain range of a particular agent A^j are called its neighbors. Formally, we can define a subset N^j of agents within a certain range R of A^j as follows:

$$N^j = A^k | A^k \neq A^j \wedge dist(A^k, A^j) < R \tag{16.3}$$

where $dist(A^k, A^j)$ is generally the Euclidean distance between two agents. Different ranges may be used to calculate cohesive, alignment and separation forces, or other factors such as the communication range of a *boid*. The average position $\mathbf{c}_t^j$ of agents within range R_c of A^j is calculated as:

$$\mathbf{c}_t^j = \frac{\sum_k x_t^k}{|(N_c)_t^j|} \tag{16.4}$$

The vector in the direction of this average position is calculated as:

$$c_t^j = \mathbf{c}_t^j - x_t^j \tag{16.5}$$

Similarly, we can calculate the average position $\mathbf{s}_t^j$ of agents within range R_s of A_j as:

$$\mathbf{s}_t^j = \frac{\sum_k x_t^k}{|(N_s)_t^j|} \tag{16.6}$$

The vector away from this position is calculated as:

$$s_t^j = x_t^j - \mathbf{s}_t^j \tag{16.7}$$

Finally, the vector a_t^j in the average direction of agents within range R_a of A^j, is calculated by the sum:

$$a_t^j = \frac{\sum_k v_t^k}{|(N_a)_t^j|} \tag{16.8}$$

The basic *boid* algorithm does not incorporate mechanisms for limiting velocity, preventing a *boid* from exiting some predefined area or to permitting a boid to avoid an obstacle. Likewise, it does not include mechanisms for goal-directed behavior. However, these have been modelled in other swarm algorithm variants. One example of an update that includes forces in the direction of a goal is:

$$v(t+1)^j = W_d v_t^j + c_1 r_1 (p_t^j - x_t^j) + c_2 r_2 (g_t - x_t^j) \tag{16.9}$$

Equation 16.9 is, in fact, the particle swarm optimization (PSO) update [9]. The terms $(p_t^j - x_t^j)$ and $(g_t - x_t^j)$ are forces in the direction of goals G^p and G^g, which have positions p_t^j and g_t respectively. G^p is defined as a goal to reach an agent's personal best or 'fittest' position found so far. G^g is defined as a goal to reach the globally fittest position found so far by all swarm members. r_1 and r_2 are numbers selected from a uniform distribution between 0 and 1. c_1 and c_2 are acceleration coefficients. Parameter values for W_d, c_1 and c_2 have been experimentally derived by Eberhart and Shi [10].

We now consider three algorithms for intrinsically motivated swarms, using the notion introduced above.

16.2.1 Crowds of Motivated Agents

Algorithm 1 models motivation as rules for the application of forces for intrinsic motivation in a *boids* framework. Various intrinsic motivations have been considered for use in swarms, including novelty [21], curiosity [37], achievement, affiliation and power motivation [14, 28]. Algorithm 1 introduces a simple form of motivation as an optimally motivating incentive (OMI), Ω^j [28]. This simple representation of motivation stipulates an incentive values that the agent finds maximally motivating. Other incentives are less motivating, with motivation inversely proportional to the difference between a goal's incentive $I(G)$ and the agent's OMI. That is:

$$M^j(G) = I^{max} - |I(G) - \Omega^j| \tag{16.10}$$

where I^{max} is the maximum available incentive. Using this approach power, affiliation, achievement or curiosity motivated agents can be defined as follows:

- **Power motivated:** power motivated individuals seek to control the resources or reinforcers of others. Thus, they tend to exhibit a preference for high-incentive goals. In the model above, power-motivated agents will have values for Ω^j that fall in the upper third of the range $[I^{min}, I^{max}]$ [28].
- **Affiliation motivated:** affiliation motivated individuals seek to avoid conflict and thus often exhibit preferences for low-incentive goals (that are not desirable to others). In the model above, affiliation-motivated agents will have values for Ω^j that fall in the lower third of the range $[I^{min}, I^{max}]$ [28].
- **Achievement motivated:** achievement motivated individuals prefer goals with a moderate probability of success. They may make a simplifying assumption that this is implied by moderate incentive. Thus, in the model above, achievement-motivated agents will have values for Ω^j that fall in the middle third of the range $[I^{min}, I^{max}]$ [28].
- **Curiosity motivated:** curious agents prefer to approach goals that are 'similar-yet-different' to goals they have encountered before. In this one-dimensional model where the only attribute of a goal is its incentive, curious agents will prefer incentives that are moderately different to previously encountered incentives and that they have not encountered recently.

It should be noted that the definitions above are one dimensional, incentive-based definitions of power, affiliation, achievement and curiosity. More complex/expressive definitions exist, both in motivation theory [18] and in the literature of computational motivation [28, 29, 38, 41, 42]. Some of the latter are discussed later in this chapter, as well as Sects. 3.7 and 14.5 of this book. The advantage of the one-dimensional models discussed here is that they are computationally inexpensive, even in large numbers of agents.

The remainder of the algorithm proceeds as follows: Each agent in the swarm is initialized with an OMI, Ω^j (line 1) [30]. At each time step, each agent senses the local state of its environment (line 4), including the features described above for position, velocity and neighbors within different ranges. Each agent then constructs a set G_t^j of highly motivating goals that conform to a condition on the current state (line 5). For example, the condition might concern proximity to a goal and level of motivation:

$$G_t^j = G^i | dist(g_t^i, x_t^j) < R_m \wedge M^j(G) > M \tag{16.11}$$

R_m is the range within which goals are considered and M is a motivation threshold. A force in the direction of each goal is included in the update equation for the agent (line 7) as follows:

$$v_{(t+1)}^j = W_d v_t^j + W_c c_t^j + W_a a_t^j + W_s s_t^j + W_m \sum_i (g_t^i - x_t^j) \tag{16.12}$$

Finally all agents are moved to their new positions (line 8). Algorithm 1 assumes that all goals and their locations are known by all agents, and that goals are generated

Algorithm 1 A swarm of motivated agents. Adapted from [28].

1: Initialise n and a society A of n agents with position, velocity, weights, ranges and optimally motivating incentive Ω^j.
2: **for** each time t **do**
3: **for** each agent A^j **do**
4: Sense the current local state $< x_t^j, v_t^j, (N_c^j)_t, (N_s^j)_t, (N_a^j)_t >$
5: Construct goal set G_t^j according to Eq. 16.11.
6: Compute $(g_t^i - x_t^j)$ for all G^i
7: Sum all forces on agent A^j using Eq. 16.12.
8: Move all agents to new positions according to Eq. 16.2.

by an entity external to the swarm. The next section considers an algorithm in which the swarm itself generates goals dynamically, while agents are exploring.

16.2.2 Motivated Particle Swarm Optimization for Adaptive Task Allocation

Another approach to a motivated swarm is to integrate intrinsic motivation with PSO for the purpose of adaptive task allocation [21]. Intrinsically motivated PSO (MPSO) can be used for search and allocation of resources to tasks, when the nature of the target task is not well understood in advance, or can change over time.

This algorithm has two parts: the first for motivation and the second for PSO as shown in Fig. 16.1. The input to the motivation component is spatially mapped sensor data $p_t(x)$ where x specifies the location from which the data were collected as a Cartesian coordinate and t is the time at which the data were collected. It is assumed that a stream of this data is input to the system. When data are collected at more than one location at time t, individual data points are denoted $p_t(x_\tau)$. The output of the motivation component, and input to the PSO, is a fitness function $F_t(x)$ as shown in Fig. 16.1.

We denote M_τ the motivation value of $p_t(x_\tau)$. In this algorithm, M_τ is assumed to be binary, with 1 denoting a motivating stimulus and 0 denoting a non-motivating stimulus. M_τ is computed by thresholding models of motivation that return a continuous value. Four such models were described above. Another example is and arousal-based model of curiosity using a novelty function, as described by Klyne and Merrick [21] and illustrated in Fig. 16.2.

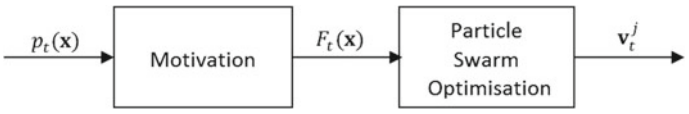

Fig. 16.1 Motivated particle swarm optimization. Image adapted from [21]

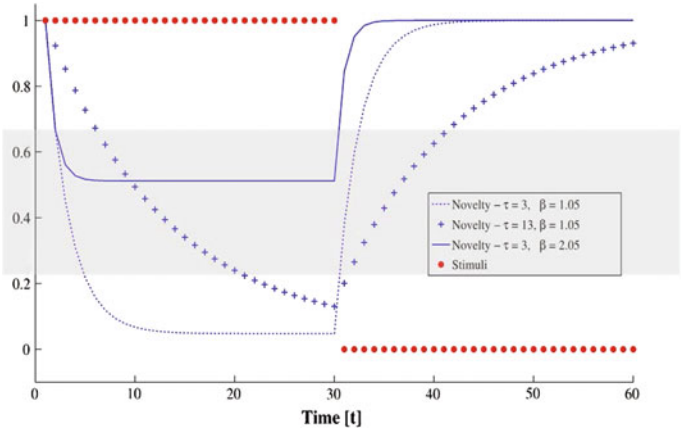

Fig. 16.2 Novelty is computed as a function of the time that a stimulus has been firing or not firing. Motivation is a binary value determined as a threshold on novelty. Motivation is 1 in the shaded area, and 0 otherwise

In this model, potential goals have multiple attributes and are represented as vectors. These vectors are clustered using an unsupervised learning algorithm such as a self-organising map (SOM), k-means clustering, adaptive resonance theory (ART) network or simplified ART network.

Neurons or cluster centres from the unsupervised learning algorithm have associated habituating units that compute novelty as shown in Fig. 16.2. The dotted series in Fig. 16.2 illustrates the activation value of a neuron that fires repeatedly (30 times), then does not fire (30 times). The solid, dashed and + series are examples of different novelty curves calculated using Stanley's model [40]. $M_\tau = 1$ when novelty is in the moderate range shown in grey in Fig. 16.2 and zero otherwise. Because novelty is influenced by the agent's experiences, as stored in their unsupervised learning component, different agents may compute different novelty values for a given stimulus because their experiences are different.

M_τ is a parameter of the fitness function, which is defined using an intensity landscape, as follows:

$$I_\tau(x) = \frac{M_\tau}{(1 + \gamma(\sum_{(y=1)}^{Y}(x^y - x_\tau^y)^2))} \tag{16.13}$$

This function forms a graduated peak with a maximum at the coordinate x_τ. The range of x is the range of the problem space. y is the counter for dimensions of the problem space. M_τ controls the maximum height of a peak on the fitness function. γ controls the gradient of a peak. Lower values make the gradient gentler. The fitness function itself is then constructed by summing intensity functions for motivating sensor data as follows:

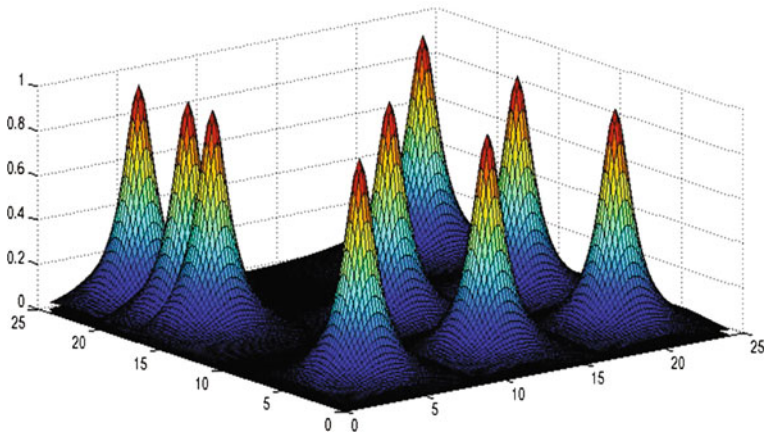

Fig. 16.3 A synthetic fitness landscape generated with Eq. 16.13 using 9 motivating points at positions (4, 4), (4, 18), (4, 19), (4, 22), (12, 4), (12, 12), (20, 4), (20, 12) and (21, 22)

$$F_t(x) = \sum_{\tau} I_\tau(x) \tag{16.14}$$

Using a sum implies that the size of an area of motivating data will influence the height of the fitness function. As an example, a synthetic fitness landscape generated using Eq. 16.13 nine motivating locations is shown in Fig. 16.3.

Algorithm 2 shows how the fitness function is incorporated with PSO in two phases: (1) a settling phase and (2) the MPSO phase. $F_0(x)$ is initialized as zero for all x. The settling phase of the algorithm (lines 3 to 8) determines the level of background noise in the fitness function by observing the environment for a fixed period T. A 'noise floor' α is then chosen by monitoring the maximum height of the generated fitness function at the end of the initialization period.

The noise floor is used to influence the inertial value of the motivated PSO phase, which commences at time $T + 1$. The motivated PSO loop (lines 10 to 21) alternates between motivation to compute an updated fitness function (lines 12 to 17) and optimization of the current fitness function (lines 18 to 21). When a motivation phase occurs, the fitness function and the values of p_t^j and g_t are reset to zero, so the swarm to diverges. The condition described in Eq. 16.15 is applied so that the inertial value W_d in the PSO update is only effective when the height of the fitness function is greater than the noise floor value established during the settling phase:

$$W_d = \begin{cases} 0.729 \text{ if } F(g_t) > \alpha \\ 0 \qquad\quad \text{otherwise} \end{cases} \tag{16.15}$$

The non-zero alternative in Eq. 16.15 is the value proposed by Eberhart and Shi [10]. This algorithm is generic enough for a range of PSO variants to be substituted at this point.

Algorithm 2 Motivated particle swarm optimisation where motivation generates a dynamic fitness function. Adapted from [28].

for each agent A^j do **do**
  Initialise with random x^j and v^j
for $t = 1$ to T do **do**
  Sense the environment
  $F_0(x) = 0$ for all values of x
  for each piece of spatially mapped senor data $p_t(x_\tau)$ do **do**
    Compute motivation M_τ for $p_t(x_\tau)$
  Generate fitness using Eq. 16.14
Set PSO noise floor $\alpha = \max_x F(x_\tau)$
for $t > T$ do **do**
  Sense the environment
  if $t \mod Z == 0$ then **then**
    $F_0(x) = 0$ for all values of x
    Reset p_t^j and g_t for all agents
    for each piece of spatially mapped senor data $p_t(x_\tau)$ do **do**
      Compute motivation M_τ for $p_t(x_\tau)$
      Generate fitness using Eq. 16.14
  else
    $F_t(x) = F_(t-1)(x)$
  Perform PSO update in Eq. 16.9
  Move all agents to new positions according to Eq. 16.2.

The motivation and PSO components may potentially run in parallel, for example on different processors, or they may be interleaved on a single centralized processor. In either case, the motivation and PSO components do not have to step at the same rate. The PSO component simply works with the current version of the fitness function available. In Algorithm 2, the ratio of motivation to optimization is controlled by the parameter Z. It is further assumed that the environment is piecewise dynamic, that is, it changes slowly enough for the PSO component to converge on an optima before its location changes again. In this algorithm all agents are motivated by a single, shared, but dynamic fitness function. The approach in the next section incorporates different models of motivation into different agents to allow the agents to exhibit different characteristics in task selection.

16.2.3 Motivated Guaranteed Convergence Particle Swarm Optimization for Exploration and Task Allocation Under Communication Constraints

This algorithm [16] uses a specific variant of the PSO algorithm, namely the guaranteed convergence particle swarm optimization (GCPSO) algorithm [33] to prevent agents from stagnation and premature convergence on suboptimal solutions., In the

case where an agents have a limited communication range, the standard PSO algorithm might deal with a situation where it is not connected with any of the agents in the population. In such a case, the agent's personal best position is equal to its own neighborhood best position. This may potentially lead to stagnation and early convergence. To deal with this problem, the idea of GCPSO, thus, involves changing the velocity update equation of the neighborhood best agent. To create Motivated Guaranteed Convergence Particle Swarm Optimization (MGCPSO), the GCPSO algorithm is combined with models of motivation. to create Motivated Guaranteed Convergence Particle Swarm Optimization (MGCPSO).

In the MGCPSO algorithm, Algorithm 3, the set of neighbors of agent A^j is defined according to Eq. 16.3 where R_{com} is the maximum communication range of the agents and $R_{com} > 0$.

As in Algorithm 2, each agent A^j is assumed to be able to remember the location of the best position it has sensed so far (personal best position), p_t^j. However, in MGCPSO, a set of potential neighborhood best positions is also maintained for the agents in the neighborhood N_{com}^j as follows:

$$G_t^j = \left\{ G | G \in N_{com}^j \wedge \left| F(g_t) = argmax F(g_t^i) \right| < \mu \right\} \tag{16.16}$$

Next, the set G_t^j is augmented with an artificial, randomly generated position in the search space, g_t^*, which results in a new set $\hat{G}_t^j$ (line 7). Then, the neighborhood best position, n_t^j, is computed using the agent's model of motivation, to selecting the most highly motivating neighborhood goal. Motivation is modelled as a profile of achievement, affiliation or power motivation (line 8). Motivation is computed as a function of incentive, where incentive itself is a function of selected situational variables. Hardhienata et al. [16] proposed that distance to goal (D) and number (a) of agents around a goal are appropriate situational variables for task allocation. Their incentive function is shown in Fig. 16.4. Three motive profiles that are a function of this incentive function are shown in Fig. 16.5. Different agents have different motive profiles and Hardhienata gives guidelines for choosing the proportions of agents with different profiles [13]. Briefly, the ideas captured by these functions are:

- **Power motivation agents:** power motivated agents are willing to take risks. In this scenario, travelling further to a goal (high D), or approach a goal that only a small number of other agents have approached (low a) constitutes a risky activity.
- **Affiliation motivated agents:** affiliation motivated agents seek out the company of other agents. Thus motivation is highest for low incentive goals, which occur for high values of a.
- **Achievement motivated:** achievement motivated individuals prefer goals with a moderate risk. In this work this is assumed to mean moderate values of D and a.

Finally, a modified version of the GCPSO velocity update is applied (line 11):

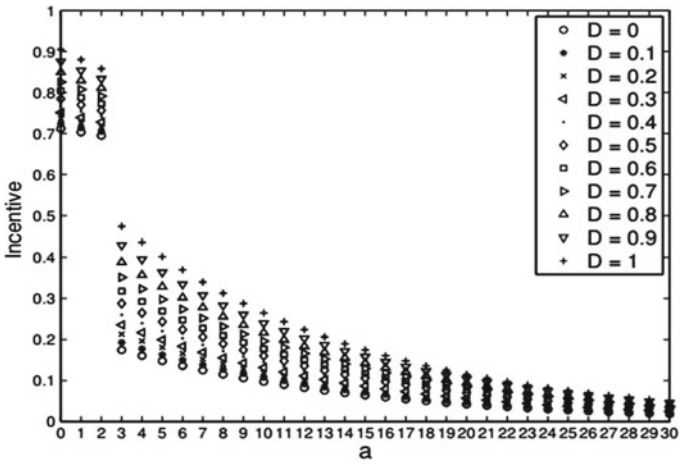

Fig. 16.4 Incentive is a function of distance to goal (D) and number of agents (a) already around the goal. Image from [16]

$$v_{(t+1)}^{j} = \begin{cases} W_d v_t^j + \lambda(1 - 2r_1) & \text{if } n_t^j = g_t^* \\ W_d v_t^j - x_t^j + n_t^j + \rho_t(1 - 2r_2) & \text{if } n_t^j = p_t^j \\ W_d \left[v_t^j v_t^j + c_1 r_3 (p_t^j - x_t^j) + c_2 r_4 (n_t^j - x_t^j) \right] & \text{Otherwise} \end{cases} \tag{16.17}$$

Algorithm 3 Motivated guaranteed convergence particle swarm optimisation where agents have different motivation functions. Adapted from [13].

1: **for** each agent A^j do **do**
2: Initialise with random x^j and v^j and various motivation constants to create agents with different profiles of achievement, affiliation and power motivation
3: **for** $t = 1$ to T do **do**
4: **for** each agent A^j do **do**
5: Compute personal best p_t^j
6: **for** each agent A^j do **do**
7: Calculate $\hat{G}_t^j$
8: Calculate maximally motivating goal(s) from $\hat{G}_t^j$
9: Select the closest goal if more than one is maximally motivating
10: **for** each agent A^j do **do**
11: Perform PSO update in Eq. 16.17
12: Move all agents to new positions according to Eq. 16.2.

ρ_t is updated based on an adaptive search procedure [33] and λ is a constant used to scale the contribution of the random search. Clerc and Kennedy [7] suggests ways to set W_d. For the first case in Eq. 16.1, the personal best position (p_t^j) and the neighborhood best (n_t^j) position are not involved. Thus, the agents will not be forced

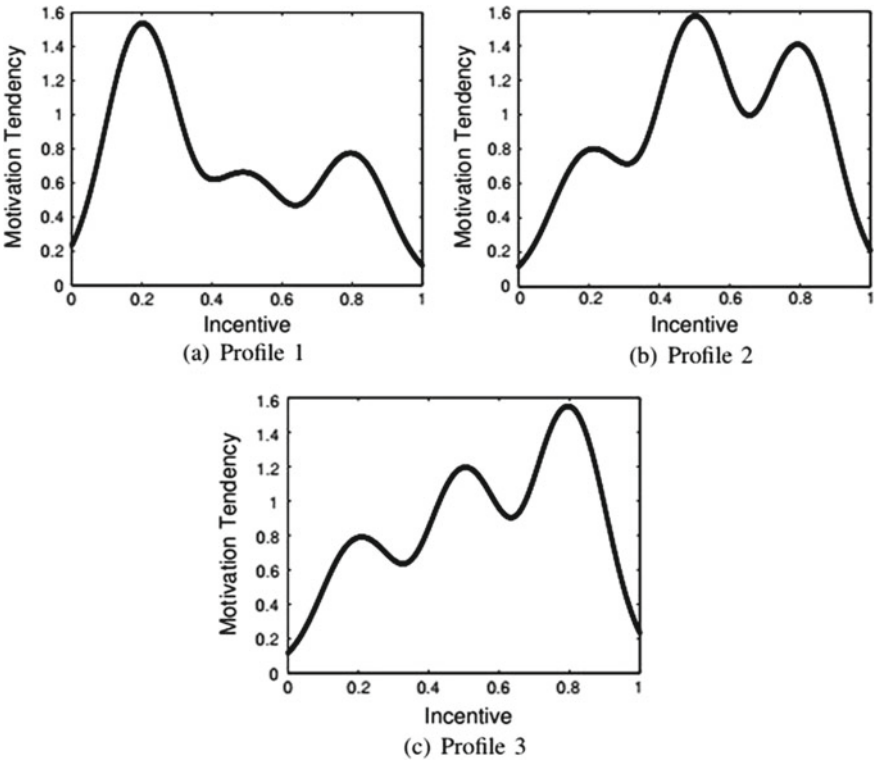

Fig. 16.5 Three profiles of motivation as a function of incentive. Image from [16] **a** a profile for agents with dominant affiliation motivation; **b** a profile for agents with dominant achievement motivation; **c** a profile for agents with dominant power motivation

to move towards their personal best and neighborhood best positions. This is done to allow the agents to perform a broader exploration of the search space. The second and third cases in Eq. 16.17, on the other hand, are based on the GCPSO algorithm. Note that compared to the standard PSO algorithm, the GCPSO algorithm differs in the case where $n_t^j = p_t^j$ to prevent stagnation.

This overview concludes our look at intrinsically motivated swarms. Other work has considered intrinsic motivation in other kinds of multi-agent systems (such as evolutionary settings [24]), but we consider this out of the scope of this paper. The next section now considers some of the advantages that have been achieved through the use of intrinsic motivation in swarm systems.

16.3 Functional Implications of Intrinsically Motivated Swarms

Empirical studies and case studies of intrinsically motivated agent swarms have revealed a number of advantages of such models in diverse applications including computer games [28], hazard detection [21] and search [13]. These advantages–including increased diversity, adaptation and capacity for exploration–are discussed in the remainder of this section. Eq. 16.14 considers more abstract implications for intrinsic motivation on the trustworthiness of autonomous systems.

16.3.1 Motivation and Diversity

A key property of motivated agents revealed particularly in Algorithm 1 and Algorithm 3 is their diversity. In Algorithm 1, agents are initialized with different OMIs so they have different preferences for incentive. In Algorithm 3, agents are embedded with different profiles of achievement, affiliation and power motivation. These profiles include more expressive models of incentive in terms of situational variables, so agents respond differently to specific aspects of their environment. Figure 16.6 demonstrates agent diversity in the Breadcrumbs game. Breadcrumbs is a simple Android game set in two rooms connected by an open doorway. Initially the characters (simple square-shaped boids in this case) are randomly distributed throughout both rooms. The rules of the game are as follows:

Aim of the game: Place up to five breadcrumbs to lure all the *boids* into one room *Instructions:* 1. Place breadcrumbs by touching the screen at the desired location 2. Once you have placed five breadcrumbs, you can continue placing breadcrumbs, but each new breadcrumb will trigger the removal of the oldest existing breadcrumb 3. Breadcrumbs are always tasty - but you don't know exactly how tasty any given crumb will be. In addition, different boids have different preferences for flavour

In Breadcrumbs power motivated agents are red, achievement motivated agents are orange and affiliation motivated agents are yellow. Breadcrumbs themselves are brown. We can see from Fig. 16.6 that agents with similar motives cluster around similar breadcrumbs. This is a demonstrator of the way motivational diversity results in behavioral diversity. Merrick [28] provides a case study comparing diversity as a result of motivation to homogeneous and random heterogeneous swarms and concludes that the systematic approach to motivation supports more predictable agent behaviour (Fig. 16.7).

Hardhienata et al. [14, 15] also report behavioral diversity as a result of motivational diversity. In their model, affiliation motivated agents tend to perform local

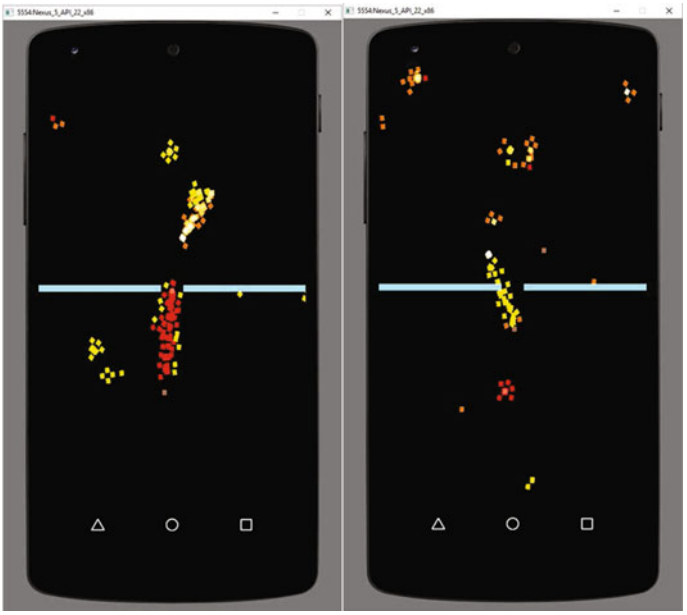

Fig. 16.6 Motivated crowds in the Breadcrumbs game

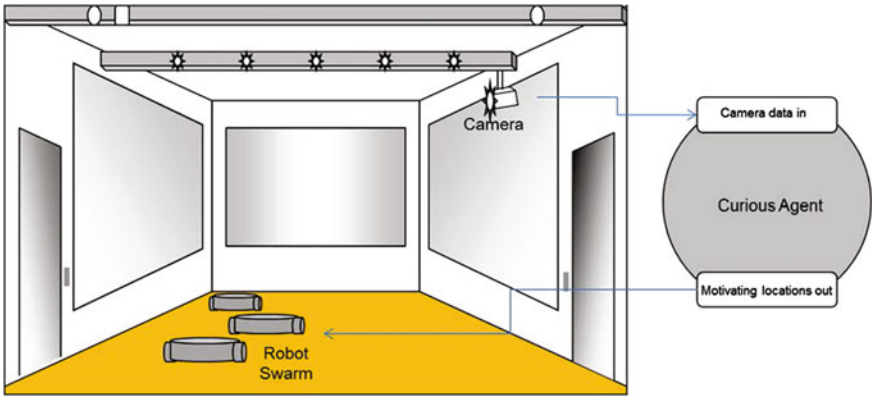

Fig. 16.7 Conceptual view of the hazard detection scenario described by Klyne and Merrick [21]. The robot swarm receives motivating locations from a centralized curious agent that analyses surveillance camera data. They constructed an image database for different floor surfaces (available at https://figshare.com/articles/Hazard_Database/3180487)

search and allocate themselves to tasks. In contrast, power-motivated agents tend to explore to find new tasks. These agents perform these characteristic behaviors more effectively in the presence of achievement-motivated agents, improving the task allocation performance of the swarm as a whole [15].

In Breadcrumbs power motivated agents are red, achievement motivated agents are orange and affiliation motivated agents are yellow. Breadcrumbs themselves are brown. We can see from Fig. 16.6 that agents with similar motives cluster around similar breadcrumbs. This is a demonstrator of the way motivational diversity results in behavioral diversity. Merrick [28] provides a case study comparing diversity as a result of motivation to homogeneous and random heterogeneous swarms and concludes that the systematic approach to motivation supports more predictable agent behaviour.

Hardhienata et al. [14, 15] also report behavioral diversity as a result of motivational diversity. In their model, affiliation motivated agents tend to perform local search and allocate themselves to tasks. In contrast, power-motivated agents tend to explore to find new tasks. These agents perform these characteristic behaviors more effectively in the presence of achievement-motivated agents, improving the task allocation performance of the swarm as a whole [15].

16.3.2 Motivation and Adaptation

Where Algorithm 1 assumes that goal locations are known by agents, and generated by an entity external to the swarm, Algorithm 2 permits the swarm to generate goals dynamically, while agents are exploring. The swarm here no longer has the diversity of Algorithm 1, as all agents share the same model of motivation, but the swarm arguably has a greater level of autonomy because it can generate its own goals.

Klyne and Merrick [21] demonstrate Algorithm 2 in a simulated hazard detection scenario. They use a swarm of agents (representing robots) to detect hazards, with the idea that the robots will either clear up, or warn passers-by of, the detected hazard. The advantage of the MPSO approach is that a strong task signature for hazards is not required. Rather hazards are identified as novel or interesting occurrences in surveillance images. A conceptual view of this setup is illustrated in Fig. 16.8 shows five images from a hazard detection scenario generated by Klyne and Merrick [21]. The first four images in Fig. 16.8 shows the fitness landscape while the algorithm is in the settling phase. The fifth image in Fig. 16.8 shows the fitness landscape and simulated robots towards the end of one of the MPSO phases. Klyne and Merrick [21] demonstrate that successive convergence and divergence of a swarm as it adapts to the introduction and removal of different hazards in each scenario.

Klyne and Merrick [21] demonstrate Algorithm 2 in a simulated hazard detection scenario. They use a swarm of agents (representing robots) to detect hazards, with the idea that the robots will either clear up, or warn passers-by of, the detected hazard. The advantage of the MPSO approach is that a strong task signature for hazards is not required. Rather hazards are identified as novel or interesting occurrences in surveillance images. A conceptual view of this setup is illustrated in Fig. 16.8. Furthermore, Fig. 16.8 shows a changing fitness landscape while the algorithm is in the settling phase. The image in the bottom row of Fig. 16.8 shows the fitness function after a hazard has been identified. Klyne and Merrick [21] demonstrate that

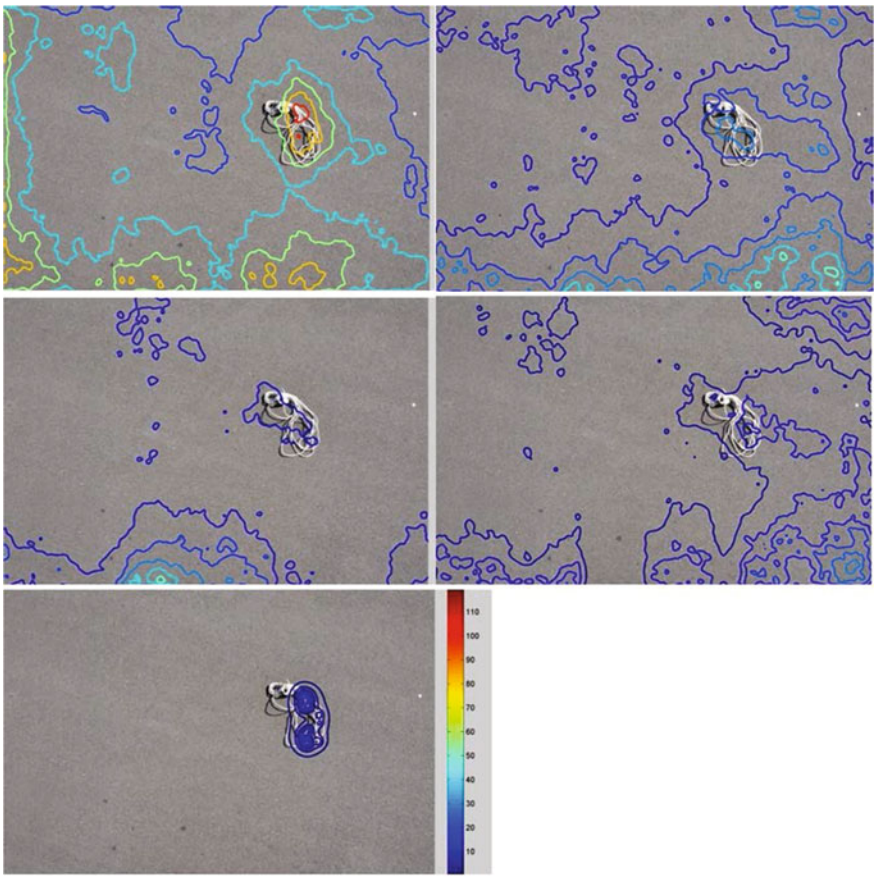

Fig. 16.8 Example of a novelty-based fitness function being generated over a bitumen surface. The first four image show the fitness function during the settling stage. The fifth image (bottom row) shows the fitness function settled over a hazard

successive convergence and divergence of a swarm as it adapts to the introduction and removal of different hazards in each scenario.

16.3.3 Motivation and Exploration

Algorithm 3 demonstrates the impact of motivation on exploration. Traditionally, simulated swarms are initialized by randomizing agents' initial positions and velocities in a defined space. However, in practice, if agents are real robots being rolled off the back of a truck or launched from a boat or aircraft, they are effectively initialized at a single point. Hardhienata et al. [16] show that algorithms such as GCPSO

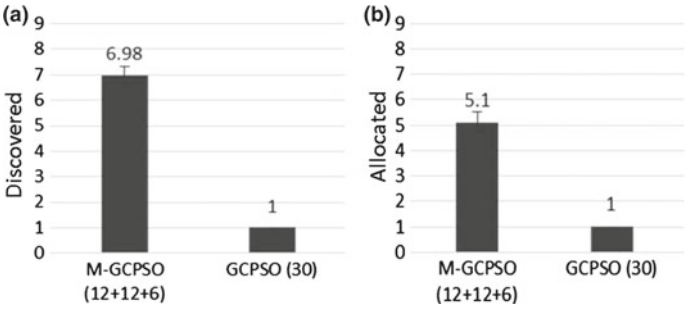

Fig. 16.9 **a** Number of goals discovered in the synthetic landscape in Fig. 16.3. when agents are initialized from a single point. **b** Number of tasks to which agents are allocated when agents are initialized from a single point

perform badly under such conditions, in terms of the number of goals they discover, and the number of goals on which they are able to converge agents. In contrast, MGCPSO significantly increases the number of discovered goals when the agents are initialized from a single point. It also increases the number of goals to which agents are allocated.

Some comparative results for the synthetic fitness function shown previously in Fig. 16.3 are shown in Fig. 16.9. These results compare 30 unmotivated agents using GCPSO to MGCPSO using agents with motive profiles 1 (12 agents), 2 (12 agents) and 3 (6 agents) shown in Fig. 16.5. Simulations also indicated this offered significant advantages when the communication of agents is limited. This is because agents can pursue goals relatively independently using their intrinsic motivation when they are not in contact with large numbers of swarm-mates.

16.4 Implications of Motivation on Trust

In the previous section we discussed how swarms of agents incorporating models of curiosity, achievement, affiliation and power can achieve diversity, adaptive behavior, and improve exploratory capabilities. All of these properties are aspects of autonomy. This then raises the question of how changes in these properties can affect trustworthiness. Trust itself is a multifaceted concept, including properties such as reliability, privacy, security, safety, complexity, risk, and free will [3]. We now consider how the properties of motivated agents considered in Sect. 16.3 might influence the perception of trust in relation to these properties.

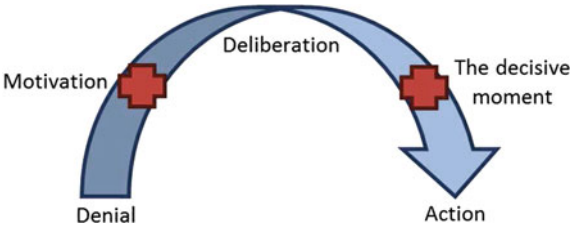

Fig. 16.10 In humans, motivation is understood to play an important role in the 'survival arc' [36], moving an actor from the denial phase through to deliberation that can result in action

16.4.1 Implications for Reliability

Reliability refers to consistency of actions, values and objectives and stability of performance during the lifetime of a trusting relationship [3]. In the context of a motivated swarm, the diversity of individuals in a swarm may be a double edged sword. At the macro level, we have seen that diversity as a result of motivation can offer improvements to factors such discovery of goals and convergence on goals [14]. Literature on human disaster survival–perhaps the ultimate demonstration of reliability–places motivation at a critical juncture of the 'survival arc' [36] (see Fig. 16.10). The survival arc has three phases: (1) denial, where the actor refuses to acknowledge abnormality in their situation; (2) deliberation: which includes milling and information gathering; and (3) action. Motivation is required to move an actor from the denial phase through to the deliberation phase before action can occur. Computational motivation has the same positive potential in artificial systems.

However, at the level of the individual, greater variability in performance is introduced. Different agents, when they encounter the same situation, will act differently as a result of their motives or experiences. Unless these internal differences are transparent to a human collaborator, there may be a perception that there is less consistency of action between individuals. Existing work has found that such performance based factors play a key role in trust development between humans and robots [12, 44].

Adaptation of a swarm also has implications for reliability. In the case where a swarm can generate its own goals, we have seen that this can have a positive impact on stability of performance because the swarm can adapt in the presence of novel hazards [21]. However, once again, there may also be negative implications for trust if there is a perception that the agent can have changing objectives (and control of its own changing objectives) during the course of its life.

One mitigation technique to deal with the impact of diversity and change in humans is offered to us by the literature on reputation [20]. Reputation models permit users ('witnesses') to rate trustees, whether human or software (intelligent or otherwise). This information can be used by others to determine whether they also should trust in the specific trustee.

While the examples discussed above give us some insight on how motivation may affect the reliability aspect of trust, there is currently very little, if any, work that actually incorporates both computational models of motivation and computational models of trust. We thus conclude this section with a number of thoughts on how

specific models of motivation may impact trustworthiness. In Sect. 16.2.3 we saw that one characteristic of power-motivated agents is an increased inclination higher risk behavior. While risk-taking behavior can have the advantage of high payoff, in situations positive return does not eventuate, this could contribute to a perception of unreliable behavior or lack of trustworthiness. Likewise, agents with embedded models of curiosity may divert from an established behavioral pattern to satisfy their need for novelty. This also has potential to contribute to a perception of reduced reliability if it does not result in any advantage such as a novel discovery or process improvement. At the other end of the spectrum, achievement-motivated agents are moderate risk-takers and seek mastery of their environment and high performance. These characteristics are well suited to reliable performance. As such, a heterogeneous society of agents with different motive profiles may be best able to harness the advantages of computational motivation while maintaining trust.

16.5 Implications for Privacy and Security

Privacy and security are related, although distinct concepts. When a trustor trusts a trustee, the trusting relationship may involve transfer of data. Any misuse of this data outside terms of the trusting contract is a breach of privacy [3]. Security has broader connotations and, while including confidentiality, also concerns the integrity and continued availability of data.

While motivated agents have not been widely examined in the context of privacy and security, some of the reported results with motivated swarms have interesting implications in this regard. Hardhienata et al. [14] presented evidence that significant performance advantages can be achieved by motivated swarms when the communication of agents is limited. This is because agents can pursue goals relatively independently using their intrinsic motivation when they are not in contact with large numbers of swarm-mates. A smaller communication radius has the potential to make a network more difficult to detect, and thus offer a security advantage in a contested environment.

As we noted in our discussion of motivation and reliability, in the case of motivation and security (or at least a lowered communication requirement) a heterogeneous society of motivated agents is best able to achieve this [14].

16.5.1 Implications for Safety

Traditional safety-critical software verification requires that every condition of every branch of software is tested and that every line of code and test can be traced back to the software's requirements [19]. By this definition, it appears that motivated agent technologies should be suitable for use in safety-critical situations. However, in systems with the capacity for learning, where behavior is influenced by experiences

and where the breadth of possible experience cannot be known in advance, the traditional definition of safety-critical verification falls short. Because the data input to the motivated agent will influence its emergent behavior, and because this data cannot be predicted in advance, it is difficult to test for all possible outcomes/behaviors.

Again noting that there is currently very little, if any, work that actually incorporates both computational models of motivation and computational models of trust, we thus conclude this section with a number of thoughts on how specific models of motivation may impact trustworthiness. As we noted earlier, power-motivated agents are characterized by an increased tendency for risk-taking and resource controlling behavior. Risk-taking behavior that does not result in positive payoff may, as a consequence, impact safety. This may in turn have a negative impact on trustworthiness. Likewise, resource controlling behavior can lead an agent into situations of conflict, which may also impact safety aspects of trust. In natural systems, power-motivation is understood to be tempered by affiliation motivation, which balances resource controlling preferences with relationship building behaviors. It may be that future artificial systems will also benefit from embedded motive profiles, rather than individual motives which has been the existing research focus.

16.6 Implications of Complexity

Trust is a form of educated delegation that a trustor may enter to manage some level of complexity [3]. A trustor delegates to a trustee when there is a benefit for the trustor in trusting rather than performing the job themselves. That is, when delegation reduces some form of complexity. Examples of complexity include technical complexity associated with performing the task, time pressure or the increase in mental and cognitive complexity if the trustor chooses to perform the task themselves. As the level of complexity increases, the degree with which a trustor trusts a trustee increases. In this context, the implications of motivation on trust are tied closely to the situation in which motivated agents are given trust. Self-motivated agents are specifically designed for complex or dynamic environments where system designers cannot predict in advance all the goals the agent may need to address. According to the definition above, such environments will require a high level of trust to be placed in motivated agents.

16.7 Implications for Risk

A trusting decision involves a level of uncertainty associated with the possibility that the trustee will breach trust. A rational definition of risk might look like [5, 8]:

$$Risk = Probability \times Consequence$$

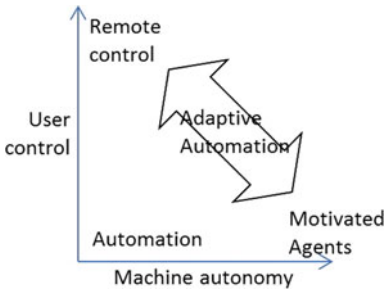

Fig. 16.11 Motivated agents in the spectrum of human control and machine autonomy

Probability refers to the probability of the given risk occurring and Consequence refers to the cost of the risk occurring. However, for humans, perception of risk, especially under pressure, may not adhere to this rational definition. An irrational component of risk, that changes the way risk is perceived, is dread [11]. The influence of dread on risk has been modelled in various ways, including as a dimension of risk [39] and as a multiplier of risk [36].

Dread represents human 'evolutionary fears', hopes, prejudices and biases. Dread itself can be represented as [36] in terms of uncontrollability, unfamiliarity, imaginability, suffering, scale of destruction and unfairness. That is, humans perceive higher risk in situations that are uncontrollable or unfamiliar, where they can easily imagine the consequences of failure, where failure will result in suffering on a large scale or over a long time, or where the situation is perceived to be undeserved. This perceived or subjective value of risk may not agree with statistical or objective values of risk.

Suppose we look at the conceptual space represented in Fig. 16.11 through the lens of dread. Figure 16.11 places different types of automations and autonomous agents on axes of 'machine autonomy' and 'user control'. We can see that motivated agents sit at the extreme low end of user control (which increases dread). Autonomous systems such as robots are also still a relatively unfamiliar technology (which increases dread) and popular media such as the Terminator series of movies aids the imaginability of disaster scenarios involving such technologies (again increasing dread). In summary, while we have described documented advantages of incorporating motivation in artificial systems, human perception of the risk associated with such systems, in particular influenced by dread, may still impact perception of their trustworthiness.

If we move to the lower level of examining specific motives with respect to risk, then we have seen that certain dominant motives will result in a stronger preference for risk-taking behavior than others. Power motivation in particular can be characterised by a preference for risk taking behavior, while affiliation motivated individuals tend to avoid such behavior.

16.7.1 Implications for Free Will

Free will is the ability of the actor to make a decision within a bounded space autonomously and at its own discretion [3]. The space may be bounded by social ties,

social rules and norms, and interdependencies among actors in terms of resources and objectives. In other words, forced trust cannot be construed as trust. In this sense, the existence of alternative solutions and technologies is a boon for emerging technologies such as motivated agents. Where users choose to trust these new technologies and are rewarded by greater reliability, privacy, security or safety, or reduced risk or complexity, trust will grow.

If we move to the lower level of examining specific motives with respect to risk, then we have seen that certain dominant motives will result in a stronger preference for risk-taking behavior than others. Power motivation in particular can be characterised by a preference for risk taking behavior, while affiliation motivated individuals tend to avoid such behavior.

16.8 Conclusion

In conclusion, this chapter has considered the impact of one of the emerging mechanisms for achieving autonomy–computational motivation–on the trustworthiness of autonomous systems. We considered this question in the context of intrinsically motivated agent swarms using some of the key variants of computational motivation: curiosity, novelty-seeking, achievement, affiliation and power motivation. Section. 16.2 provided an overview of the theory underlying the use of computational motivation in swarms of artificial agents, including a uniform notation for three intrinsically motivated swarm algorithms. Section. 16.4 considered the implications of motivation for the functionality of agent swarms, including diversity, adaptation and greater capacity for exploration. Finally Sect. 16.3 considered the implications of motivation on trustworthiness, both at the level of individual motives and at the level of permitting or not permitting intrinsic motivation in an artificial system.

Finally, in answer to the question framed in the title of this chapter: Computational Motivation, Autonomy and Trustworthiness: Can We Have It All? we present the following thoughts:

- Initial evidence suggests that inclusion of intrinsic motivation in artificial agents is likely to impact trustworthiness, but this may be in either a positive or negative sense. We saw positive impacts on performance that may translate to impacts on reliability, but also impacts on safety or risk facets of trust that may be perceived as negative.
- Approaches to the inclusion of motivation in artificial systems that may further modify the impact of motivation on trust include (1) which motives are used in artificial agents, and how or whether multiple motives are combined in a single agent or (2) in societies of agents whether individuals are homogeneous or heterogeneous.
- Motivated agent technology must remain transparent to combat factors such as dread and its associated impact on trustworthiness.

References

1. *Autonomous Systems: Social, Legal and Ethical Issues* http://www.raeng.org.uk/publications/reports/autonomous-systems-report (The Royal Academy of Engineering, London, UK, 2009)
2. H. Abbass, *Computational Red Teaming: Risk Analytics of Big-Data-to-Decisions Intelligent Systems* (Springer Verlag, Berlin, 2015)
3. H. Abbass, G. Leu, K. Merrick, A review of theoretical and practical challenges of trusted autonomy in big data. IEEE Access **4**, 2809–2830 (2016)
4. H. Abbass, E. Petraki, K. Merrick, J. Harvey, M. Barlow, Trusted autonomy and cognitive cyber symbiosis: open challenges. Cogn. Comput. **8**(3), 385–408 (2016)
5. G. Ballard, *Industrial Risk: Safety by Design* (Wiley, Chichester, 1992)
6. C.E. Billings, *Aviation Automation: The Search for a Human-Centred Approach* (Lawrence Erlbaum Associates, Mahwah, NJ, 1997)
7. M. Clerc, J. Kennedy, The particle swarm - explosion, stability and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6**(1), 58–73 (2002)
8. M. Douglas, *Risk and Blame: Essays in Cultural Theory* (Routledge, London, 1992)
9. R. Eberhart, J. Kennedy, A new optimiser using particle swarm theory (1995)
10. R. Eberhart, Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimisation (2000)
11. R. Gregory, R. Mendelsohn, Perceived risk, dread and benefits. Risk Anal. **13**(3), 259–264 (1993)
12. P.A. Hancock, D.R. Billings, K.E. Schaefer, J.Y.C. Chen, E. de Visser, R. Parasuraman, A meta-analysis of factors affecting trust in human-robot interaction. Hum. Factors **53**, 517–527 (2011)
13. M. Hardhienata, *Models of Motivation for Particle Swarm Optimization with Application to Task Allocation in Multi-Agent Systems*. PhD thesis (2015)
14. M. Hardhienata, K. Merrick, V. Ougrinovski, *Task allocation in multi-agent systems using models of motivation and leadership* (Presented at the IEEE conference on evolutionary computation, Brisbane, Australia, 2012), pp. 86–93
15. M. Hardhienata, K. Merrick, V. Ugrinovskii, Effective motive profiles and swarm compositions for motivated particle swarm optimisation applied to task allocation. Presented at the IEEE symposium series on computational intelligence, symposium on computational intelligence for human-like intelligence, (2014)
16. M. Hardhienata, V. Ugrinovskii, K. Merrick, Task allocation under communication constraints using motivated particle swarm optimization. Presented at the IEEE congress on evolutionary computation, CEC **2014**, 3135–3142 (2014)
17. R. Hardin, The street-level epistemology of trust. Polit. Soc. **21**, 505–529 (1993)
18. J. Heckhausen, H. Heckhausen, *Motivation and Action* (Cambridge University Press, New York, NY, 2010)
19. J. Hinchman, M. Clark, J. Hoffman, B. Hulbert, C. Snyder, Towards safety assurance of trusted autonomy in air force flight critical systems (2012)
20. A. Josang, R. Ismail, C. Boyd, A survey on trust and reputation systems for online service provision. Decis. Support Syst. **43**(2), 618–644 (2007)
21. A. Klyne, K. Merrick, Intrinsically motivated particle swarm optimisation applied to task allocation for workplace hazard detection. Adapt. Behav. **24**(4), 219–236 (2016)
22. R.M. Kramer, Trust and distrust in organisations: emerging perspectives, enduring questions. Annu. Rev. Psychol. **50**, 569–598 (1999)
23. A. Lacher, *Research Challenges Associated with Unmanned Aircraft Systems Airspace Integration*. MITRE Corporation (2012)
24. J. Lehman, K. Stanley, Abandoning objectives: evolution through search for novelty alone. Evol. Comput. **19**(2), 189–223 (2011)
25. J.C.R. Licklider, Man-computer symbiosis. IRE Trans. Hum. Factors Electron. **1**, 4–11 (1960)

26. A.R. Lomuscio, *Trusted Autonomous Systems* (Engineering and Physical Sciences Research Council, Department of Computing, Imperial College London, 2011–2016)
27. D. McKnight, N. Chervany, *Trust and Distrust Definitions: One Bite at a Time* (Springer, Berlin, 2001), pp. 27–54
28. K. Merrick, *Computational Motivation for Game-Playing Agents* (Springer, Berlin-Heidelberg, 2016)
29. K. Merrick, K. Shafi, Achievement, affiliation and power: motive profiles for artificial agents. Adapt. Behav. **19**(1), 40–62 (2011)
30. K. Merrick, K. Shafi, A game theoretic framework for incentive-based models of intrinsic motivation in artificial systems. *Frontiers in Cognitive Science, Special Issue on Intrinsic Motivations and Open-Ended Development in Animals, Humans and Robots* p. 4 (2013)
31. P.-Y. Oudeyer, F. Kaplan, V.V. Hafner, Intrinsic motivation systems for autonomous mental development. IEEE Trans. Evol. Comput. **11**(2), 265–286 (2007)
32. R. Parasuraman, T. Bahri, J.E. Deaton, J.G. Morrison, M. Barnes, *Theory and Design of Adaptive Automation in Aviation Systems* (Technical report, Naval Air Development Centre, 1992)
33. E. Peer, F. van den Bergh, A. Engelbrecht, Using neighbourhoods with the guaranteed convergence pso (2003)
34. E. Petraki, H. Abbass, On trust and influence: a computational red teaming game theoretic perspective (2014)
35. C.W. Reynolds, Flocks, herds and schools: a distributed behavioral model. Comput. Graphics (SIGGRAPH 87 Conf. Proc.) **21**(4), 25–34 (1987)
36. A. Ripley, *The Unthinkable: Who Survives When Disaster Strikes and Why* (Three Rivers Press, New York, 2009)
37. R. Saunders, J.S. Gero, Curious agents and situated design evaluations. Artif. Intell. Eng. Des. Anal. Manuf. **18**(2), 153–161 (2004)
38. C. Simkins, C. Isbell, N. Marquez, Deriving behavior from personality: a reinforcement learning approach, in *International Conference on Cognitive Modelling*, pp. 229–234
39. P. Slovic, Perception of risk. Science **236**(4799), 280–285 (1987)
40. J. Stanley, Computer simulation of a model of habituation. Nature **261**, 146–148 (1976)
41. R. Sun, Motivational representations within a computational cognitive architecture. Cogn. Comput. **1**, 91–103 (2009)
42. R. Sun, P. Fleischer, A cognitive and social simulation of tribal survival strategies: the importance of cognitive and motivational factors. J. Cogn. Cult. **12**, 287–321 (2012)
43. J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, E. Thelen, Artificial intelligence: autonomous mental development by robots and animals. Science **291**, 599–600 (2001)
44. R. Yagoda, D. Gillan, You want me to trust a robot? the development of a human robot interaction trust scale. Int. J. Soc. Robot. **4**, 235–248 (2012)