# Chapter2: Literature Review

Swarm Intelligence (SI) is defined as the emergent complex behaviours of autonomous agents who individually exhibit low level intelligence (*Walters, 2011)*. It is an area of artificial intelligence (AI) in which nature-inspired algorithms are created and applied to various optimisation problems, usually those of a non-deterministic polynomial time hardness (NP-hardness). This implies that an algorithm for solving one of these problems can effectively be translated into a form to solve any non-deterministic polynomial time problem (NP-problem).  
  
Much of the research in SI has taken place over the last two decades, as interest in search optimisation, load balancing, shortest path optimisation and other areas has increased. As such, much of the research is highly theoretical with some implementations as computational experiments.

This study aims to explore the implementable practicality of SI with a focus on algorithms inspired by social insect behaviours, specifically focusing on the natural shortest path abilities demonstrated by the ant and possible applications inspired technologies.

## 2.1 Metaheuristics

To understand the potential of SI it is important to first define the overall function of a meta-heuristic algorithm, where they fit and the advantages which they provide.

A meta-heuristic functions at a high level, they are designed to search for an acceptable solution within a search space without the need for entirely complete or accurate datasets (*Glover & Kochenberger, 2003)*, as such, they have become useful for confronting complex combinatorial problems.   
  
In comparison with alternative algorithms, meta-heuristics do not guarantee a solution which reaches its global optimum (*Jaszkiewicz & Branke, 2008)*. Many of the algorithms do however implement a method of stochastic optimisation, defining a global minimum/maximum of statistical functions. As such, they work on solving problems by iteratively improving upon candidate solutions (*Ombach, 2015).*

### 2.1.1 Functionality

Meta-heuristics are problem independent, they provide a set of strategies to develop heuristic optimisation algorithms, functioning as a framework from which to design algorithmic solutions (*Sörensen and Glover, 2013*). While there is an inherent absence of a universal framework for implementing meta-heuristics, this has led to a divergence of strategies in the industry including, but not limited to:

* Genetic Algorithms (GA’S)
* Swarm Intelligence (SI)
* Simulated Annealing (SA)
* Neural Networks (NN’s)

### 2.2 Swarm Intelligence

Swarm Intelligence (SI) is an emerging field in meta-heuristics, introduced via cellular robotic systems in 1989 by Beni and Wang (*Garg, Gill, Rathi & Amardeep, 2009*). These cellular systems consist of a finite number of autonomous agents operating on a finite nth dimensional space with limited local communicative capabilities to complete some predefined global task (*Wang & Beni, 1989*).

Bonabeau, Dorigo and Theraulaz argue that the term swarm intelligence creates unnecessary restrictions and redefine the definition in “*Swarm Intelligence, 1999”* to encapsulate any algorithmic designs or problem-solving devices which have been inspired by the social behaviours of insect colonies (*Bonabeau, Theraulaz, Deneuborg, Aron & Camazine, 1997).* This argument is supported by Flake (*1999)* who defines SI as that of the emergent complex behaviours which occur from groups of simple autonomous agents.

SI operates as a collective behaviour of decentralised self-organisation, as such there are no leaders and no global plan to follow. As humans we tend to imagine swarms of bees, flies or other insects but, the definition applies to a wide variety of natural behaviours in many differing organisms: chickens, flocks of birds, schools of fish, colonies of ants, cells.

From these instincts arise some interesting complex behaviours; how a flock of birds might avoid a predator, or the formations they may take while migrating, or schools of fish avoiding a shark. These emergent behaviours occur as individual agents without supervision, each agent has a probabilistic behaviour and a local perception of their environment. This system is governed by local rules and the interactions shared by the agents, leading to displays of complex behavioural patterns (*Karaboga & Akay 2009*). While these behaviours appear deliberate, the global outcome of localised actions is unknown to each agent individually.

Swarm Intelligence systems (SIS) utilise this local interaction to develop solutions; for instance; a school of fish must stay close together yet avoid collisions and possibly avoid large predators whilst maintaining a safe distance to their collective. The resulting behaviours of this collective intelligence can be used in various optimisation problems (e.g. routing, networking, load balancing, military applications).

The emergent behaviours can be attributed to four characteristics (*Bonabeau, Dorigo, & Theraulaz, 1999*):

* Positive feedback – allowing for positive reinforcement
* Negative feedback – a counterbalance to the positive feedback
* Probabilistic fluctuations – necessary for a creative approach to a problem
* Multiple Interaction – agents must share information with other agents to spread knowledge throughout the collective.

#### 2.2.1 Positive Feedback

Positive feedback is necessary to promote the occurrence of these collective behaviours. It is used to reinforce fragments of good solutions that contribute to the overall goal. Positive feedback is usually accomplished via creating a positive feedback loop; these loops move the system away from equilibrium increasing entropy. In social insects this feedback loop is accomplished via the use of pheromones. For example; ants begin their foraging behaviours with random movements outside their nest as they seek out a food source. As they wander they deposit trails of pheromones, once an ant has found food it returns to the nest depositing more pheromone along its trail. This autocatalytic process attracts more ants to the trail who in turn strengthen the trail further (*Bundgaard, Damgaard, Dacara & Winther, 2002*).

#### 2.2.2 Negative Feedback

Negative feedback acts as a counterweight to positive feedback, this is often modelled as the evaporation of pheromone along a path. Ants must be continuously moving over a path to reinitiate the pheromone as it evaporates, else the trail evaporates over time (*Bundgaard, Damgaard, Dacara & Winther, 2002*).   
  
Negative feedback prevents the ants becoming stuck in a cycle, since bad solutions can be forgotten. Exhausted supplies will not be visited and longer trails are more likely to be forgotten in favour of shorter ones, due to the frequency at which ants are able to pass over these paths relative to the time it takes to return via the same path.

#### 2.2.3 Probabilistic Fluctuations

Self-organising behaviours are reliant on random action to be taken. Creativity and innovation evolve from random fluctuations, in the same way as the earth came to be, smaller solutions develop in the same way (*Bonabeau, Dorigo, & Theraulaz, 1999)*. It is these random fluctuations in which ants may discover paths to food, random walkers may stumble upon paths which are closer than those already known.

#### 2.3.4 Multiple Interaction

All self-organisation activities require a form of shared interaction. Agents must be able to make use of their own experiences and those of other agents’ activities, networks of trails are able to be exploited by all agents, allowing for the passing of locational information to each member of the nest. Ants use of pheromones is a form of stigmergic communication, in which they can alter their environment to relay data to other ants which then respond to the change (e.g. by following a stronger scented path) (*Bundgaard, Damgaard, Dacara & Winther, 2002*). .

## 2.4 Self-Organising Systems

Systems which exhibit organised behavioural patterns are often assumed to occur by intelligent design. In such a system the actions performed occur in an order such that the global intelligent behaviour appears premeditated. This is however not the case, this emergent coordination can instead be the product of local interactions between individual agents of low individual intelligence (*Flake, 1999*). The theory of emergent behavioural organisation transpires from studies in ecology extending into computer science. The core concepts of which can be applied throughout all systems of a self-organising nature; from cells to larger organisms and even apply at a universal scale (*Renyue Cen, 2014*).   
Organisation itself is defined via measurements, the higher the entropy of a system the more information is required to describe said system.  
  
For a system to sustain itself over time it must acquire the ability to adapt to changes in its operational environment, many self-organising systems are able to remedy damage to continue operation. For instance, a phenomenon known as the *network effect* *(Dutta, 2018)* predicts that as more users join a social network the more useful that network will become up until reaching a critical mass. However, should a single user remove themselves from the network their node will soon be replaced or re-routed preserving the overall function of the social network itself. This network grows and adapts as any other self-organising system using a positive feedback loop.   
  
Diversity and innovation amongst an initial population are necessary for emergent self-organising behaviours to occur. In systems of agents with low-levels of intelligence this can be modelled by integrating inherent randomness as a part of the overall local automata.

## 2.5 Ant Colony Optimisation

Inspired by the findings of Goss (*1989)* ant colony optimisation (ACO) was first proposed by Dorigo (*1992)* originally propositioned as ant system (AS). Originally created for optimal path finding in a graph AS is based on the natural foraging behaviours of ant colonies. AS has since been redefined as the ACO meta-heuristic (*Dorigo & Caro, 1999*) and adapted to solving a range of differing problem statements.   
  
ACO is a multi-agent system best suited to discrete optimisation problems in which properties of the environment change over time, implying that the optimisation process must also be able to adapt with these changes. As there is no defined design framework, many variations of ACO have been used over the years with varying degrees of success; Dorigo and Caro rectify this unifying the design process of ACO for more simplistic implementation of the algorithm to applications (*Dorigo & Caro, 1998*).

### 2.5.1 Artificial Ants

Comparable to biological ants, artificial ants work as a collective intelligence to complete larger tasks. The individuals of the colony interact at a local level to solve a given task, modifying their environment in order to relay information via stigmergic pheromone communication. This pheromone trail is modelled via numeric information which is only obtainable at the local level. Both biological and artificial ants utilise iterative techniques for the construction of minimal cost solutions from an initial position (dependent on the algorithm this is either from a nest or a randomly generated starting position).   
  
Artificial ants build their solutions by applying positional transformations iteratively, as (unlike real ants) they exist in a discrete environment, artificial ants may exploit heuristic information along with these pheromone trails to develop efficient solutions to hard combinatorial problems. These agents also have a memory which stores the path followed by the ant.   
  
Much like certain specifies of ant more advanced editions of the ACO algorithms deposition of pheromone is dependent on the quality of the solutions, the difference being that many artificial ants only deposit pheromone upon acquiring a complete solution.   
  
Unlike biological ant’s pheromone evaporation is inherently different in artificial systems. The inclusion of pheromone evaporation is a key mechanism in preventing the algorithm becoming stuck in local optima. This evaporation allows the ant to forget previous sub-optimal paths as pheromone evaporation occurs at a swifter rate than in nature (*Rutkowski, 2005*).

### 2.5.2 Constructing Solutions

When constructing solutions, the ants follow the strongest pheromone trail. To allow the ants to deliberate and discover solutions of improved efficiency over the current best it is fundamental to the algorithm that some form of randomness be applied. The following equation denotes the probability of selecting a single component to the solution;

**Figure 1. ACO Solution Equation  
 (*Shekhawat, Poddar, Boswal 2009*)**

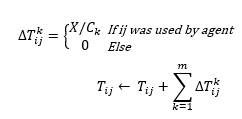
The amount of pheromone is represented via *T* with *i* and *j* being the nodes between which the ant will transpose. The parameter of which regulates the influence of the pheromone (*Tij*) is denoted by the Greek letter alpha (α) while beta (β) is representative of the heuristic value in control of the edge desirability’s influence () (*Shekawat, 2009*). The overall equation calculates the probability of which a single component within the solutions is selected.

#### 2.5.2.1 Local Pheromone Update Rule

Each time a successful solution is constructed the pheromone update function runs. This acts as the imitation of the way biological ants lay pheromones (*Dorigo, 1992*). In the artificial model, pheromone updates are replicated by varying the level of deposited pheromone on a path dependent on the overall score of the completed solution. The amount of pheromone which is deposited ensues as a result of the following equation;

**Figure 2. Pheromone Update Equation  
 (*Shekhawat, Poddar, Boswal 2009*)**

The above equation is a superior method of implementation compared to the original AS system. This is due to the inclusion of pheromone evaporation conveyed as (ρ) allowing for inefficient solutions to be “forgotten” over time (*Bundgaard, Damgaard, Dacara & Winther, 2002*). The pheromone on any particular edge is denoted as *Tij,* when multiplied with delta this gives the quantity of deposited pheromone which in turn can be determined by;



**Figure 3. Pheromone Update Equation  
 (*Shekhawat, Poddar, Boswal 2009*)**

Where (*Ck*) defines the total cost of constructing the solution. Typically, the parameter *X* is set to 1, as the parameter to adjust the deposited pheromone, however in some problem statements (such as TSP) this would be adjusted via a value such as the length of the agent’s journey.

#### 2.5.2.2 Global Pheromone Update Rule

The global rule dictates the point in which pheromone evaporates from the paths. This method is executed at the end of each iteration, established when each agent has successfully applied the local pheromone update rule (*Brownlee, 2011)*.

### 2.5.3 ACO Methodologies

Implementations of ACO have many variants, the main three of which consist of the following implementations;

#### 2.5.2.1 ACS Breakdown

ACO begins searching for possible solutions by moving through a finite sequence of neighbouring states, each move is selected via a stochastic local.

**Begin**

Initialize

**While** stop criterion not met **do**

Position each ant at starting node

**Repeat**

**For each** ant **do**

Choose next node by applying transition state rule

Apply step by step pheromone update

**End for**

**Until** every ant has built a solution

Update best solution

Apply offline pheromone update

**End While**

**End**

***Figure 4. Ant Colony System (ACS) Pseudocode.  
(Dorigo & Caro 1998)***

The algorithm begins by setting parameters and initializing pheromones. Once entering the loop, the ants are positioned at random starting nodes, another loop is then executed applying the transition rule to each ant and the local pheromone update method is called. Once every ant has built a solution the global pheromone is updated.

# Bibliography

Dorigo, M.D. & Di Caro, G.A.D.C (1998). Ant Colony Optimization. A New Meta-Heuristic, 2 , 1470-1477. Retrieved from <https://www.researchgate.net/publication/2831286_The_Ant_Colony_Optimization_Meta-Heuristic>

Goss, S.G., Aron, S.A., Deneubourg, J.L.D., & Pasteels , J.M.P. (1989). Self Organized Shortcuts of the Argentine Ant, Retrieved from <https://www.researchgate.net/publication/301232811_Self-Organized_Shortcuts_in_the_Argentine_Ant>.

Dorigo, M.D. & Stützle, T.S. (2004). Ant colony optimization, Retrieved from <https://ieeexplore.ieee.org/document/4129846>.

Bell, J.B. & Griffiths, S.E.G. (2010). Swarm Intelligence: Application of the Ant Colony Optimization Algorithm to Logistics-Oriented Vehicle Routing Problems, Retrieved from <https://www.researchgate.net/publication/230370094_Swarm_Intelligence_Application_of_the_Ant_Colony_Optimization_Algorithm_to_Logistics-Oriented_Vehicle_Routing_Problems>.

Gao, R.G. & Wu, J.W. (2015). Swarm Intelligence: Application of the Ant Colony Optimization Algorithm to Logistics-Oriented Vehicle Routing Problems, Retrieved from <https://www.mdpi.com/1999-5903/7/4/465/pdf>.

Li, K.L., Xu, G.X., Zhao, G.Z., Dong, Y.D., & Wang, D.W. (2011). Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization, Retrieved from <https://ieeexplore.ieee.org/document/6051750>.

Walters, L.P.W. (2011). Applications of Swarm intelligence, Nova Science Publishers Inc

Brownlee, J.B. (2011). Clever Algorithms: Nature-Inspired Programming Recipes. Retrieved from <http://www.cleveralgorithms.com/nature-inspired/index.html>

Bonabeau, B.E., Theraulaz, T.G., Deneubourg, J.L.D., Aron, A.S., & Camazine, C.S. (1997). Trends in Ecology and Evolution. Self-organization in social insects, 188-93. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0169534797010483>.

Angus, D.A. & Woodward, C.W. (2009). Multiple objective ant colony optimisation, 3 (1), 69-85. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0169534797010483>.

Glover, F.W.G. & Kochenberger, G.A.K. (2003). Handbook of Metaheuristics. Retrieved from <https://www.springer.com/gb/book/9780306480560>  
  
Jaszkiewicz, A.J. & Branke, J.B. (2003). Lecture Notes in Computer Science. Interactive Multiobjective Evolutionary Algorithms, 5252 , 179-193. doi: 978-3-540-88908-3.

Ombach, J.O. (2015). Lecture Notes in Computer Science. Interactive Multiobjective Evolutionary Algorithms, doi: 10.4467/20838476SI.14.001.3018.  
  
Sörensen, K.S. & Glover, F.G. (2013). Encyclopedia of Operations Research and Management Science. A History of Metaheuristics, doi: 10.1007/978-1-4419-1153-7\_1167.

Karaboga, D.K. & Akay, B.A. (2009). Artificial Intelligence Review. A History of Metaheuristics, Retrieved from <https://link.springer.com/article/10.1007/s10462-009-9127-4>.

Hinchey, M.G.H., Sterritt, R.S., & Rouff, C.R. (2009). Swarms and Swarm intelligence, 4 , . Retrieved from <http://doi.ieeecomputersociety.org/10.1109/MC.2007.144>.

Bundgaard, M.B., Damgaard, T.C.D., & Winther, W.W. (2002). Ant Routing System, Retrieved from <https://pdfs.semanticscholar.org/fecc/ba6e82f40f17831e6fd3f133d291e01f7a9d.pdf>.

Garg, A.G., Gill, P.G., Rathi, P.R., Amardeep, ., & Garg, K.K. (2009). International Journal of Recent Trends in Engineering. An Insight into Swarm Intelligence, 2 (8), Retrieved from <https://pdfs.semanticscholar.org/f3ec/1b3793146f17920d02fd7d43dc0bf946b0f0.pdf>.

Wang, J.W. & Beni, G.B. (1989). International Journal of Recent Trends in Engineering. An Insight into Swarm Intelligence, , . doi: 10.1109/ISIC.1989.238706.  
  
Flake, G.W.F. (1999). The Computational Beauty of Nature, , . doi: 10.2307/2589369.

Colorni, A.C., Dorigo, M.D., & Maniezzo, V.M. (1991). European Conference on Artificial Life,. Distributed Optimization by Ant Colonies, Retrieved from <https://www.researchgate.net/publication/216300484_Distributed_Optimization_by_Ant_Colonies>.

Cen, R.C. (2014). The Astrophysical Journal Letters. American Astronomical Society logo American Astronomical Society logo Institute of Physics logo Institute of Physics logo A publishing partnership TEMPORAL SELF-ORGANIZATION IN GALAXY FORMATION, 785 (2), Retrieved from <http://iopscience.iop.org/article/10.1088/2041-8205/785/2/L21>.

Dutta, A.D. (2018, 08 06). What Is The Network Effect? Why Is It Valuable? [Web log post]. Retrieved from <https://www.feedough.com/network-effect/>.

Rutkowski, L.R. (2005). Computational Intelligence: Methods and Techniques

Shekhawa, A.S., Poddar, P.P., & Boswal, D.B. (2009). *Ant Colony Optimization Algorithms: Introduction and Beyond.* Retrieved from <http://mat.uab.cat/~alseda/MasterOpt/ACO_Intro.pdf>.