Emergence and Exploitation of Collective Intelligence of Groups

POLITECNICO DI BARI

DEPARTMENT OF MECHANICS, MATHEMATICS AND MANAGEMENT

Ph. D. Program
in
Mechanical and Management Engineering

SSD: ING-IND/13 — MECHANICS OF MACHINES

EMERGENCE AND EXPLOITATION OF COLLECTIVE INTELLIGENCE OF GROUPS

Doctoral Thesis

Ilario De Vincenzo

Coordinator:	Prof. Ing. Giuseppe Pompeo Demelio
Supervisors:	Prof. Ing. Giuseppe Carbone
	Prof. Ing. Ilaria Giannoccaro
Referees:	Prof. Paolo Grigolini
	Prof. Mirta Galesic

XXIX CYCLE, ACCADEMIC YEARS 2014 – 2016

POLITECNICO DI BARI

DEPARTMENT OF MECHANICS, MATHEMATICS AND MANAGEMENT

Ph. D. Program
in
Mechanical and Management Engineering

SSD: ING-IND/13 — MECHANICS OF MACHINES

EMERGENCE AND EXPLOITATION OF COLLECTIVE INTELLIGENCE OF GROUPS

Doctoral Thesis

Ilario De Vincenzo

Coordinator:	Prof. Ing. Giuseppe Pompeo Demelio	_____
Supervisors:	Prof. Ing. Giuseppe Carbone	_____
	Prof. Ing. Ilaria Giannoccaro	_____
Referees:	Prof. Paolo Grigolini	
	Prof. Mirta Galesic	

XXIX CYCLE, ACCADEMIC YEARS 2014 – 2016

Ai miei genitori,
salde e profonde radici
della mia persona.
.

Acknowledgment

Non è stato semplice racchiudere in poche righe le tante emozioni di questi tre anni.
Invito per questo chi le leggerà a meditare su ogni parola impiegata, per l'elevato peso specifico che ho voluto conferirle dalla parte più profonda di me.
Ho iniziato questo percorso rinunciando, contro ogni tipo di logica altrui e per la terza volta nella mia vita, ad un'alettante proposta lavorativa. L'ho fatto perché dopo un diploma e due lauree nello stesso ambito, non mi sentivo preparato come desideravo, ero un Ingegnere per lo Stato, ma non ero abbastanza per me. Sentivo di dover crescere ancora e tanto, e da tempo avevo individuato Chi, in tutto questo, poteva aiutarmi, la persona più capace ed intelligente che abbia finora incrociato nel mio intero percorso di vita, l'unica che mi ha costantemente e da sempre stupito, il Prof. Carbone. Sin dal nostro primo incontro, ho visto in Lui quello che sarei voluto diventare. Profondo conoscitore della fisica alla base dei fenomeni, e dotato di un notevole background matematico, Giuseppe ha l'ineguagliabile capacità critica di analizzare la realtà, individuare l'essenziale e renderlo in eleganti equazioni, trasferendo così le sue intuizioni con una semplicità disarmante. In questo lo reputo irraggiungibile.
Gli chiesi di assegnarmi un tema nuovo, multidisciplinare, ambizioso, che mi avrebbe anche permesso di ampliare le mie conoscenze nelle branche della fisica e della matematica applicata, discipline che hanno sempre esercitato un fascino non indifferente nei miei confronti. Abbiamo così prima tentato la strada dell'energy harvesting, poi intrapreso quella dei sistemi complessi e della social science. Questa scelta ha condotto a non poche difficoltà, l'argomento era nuovo per tutti, richiedeva in primis un consolidamento dello stato dell'arte in un ambito che poco aveva a che fare con il nostro background ingegneristico, e nuove idee per fornire un contributo originale e rilevante. Non ho timore nel dire che ci son stati periodi neri, in cui abbiamo letteralmente brancolato nel buio.
I momenti di sconforto non sono mancati, le batoste dei primi paper rigettati su idee che noi ritenevamo meritevoli, e per me, neofita del mondo della ricerca scientifica, non è stato assolutamente semplice. In tutto questo, tuttavia, Giuseppe è sempre stato forte e saggio al mio fianco, spronandomi a non mollare, a rimboccarmi le maniche e ad imparare dalle sconfitte, rialzandosi dopo ogni caduta, perché è solo così che si diventa grandi. Questo processo ha profondamente cambiato il mio modo di essere, mi ha portato a confrontarmi

con i miei limiti, quelli che presuntuosamente prima non vedevo, ed ho riscoperto l'importanza del valore dell'umiltà, al punto che dopo tre anni, oggi posso finalmente dire di essere diventato, prima che un dottore di ricerca, un uomo ed una persona migliore.

Un doveroso ringraziamento va alla Prof.ssa Ilaria Giannoccaro, riferimento sicuro e imprescindibile, la sua analisi critica e determinata di ciò che facevamo, nonché i suoi preziosi consigli, si sono rivelati fondamentali per lo sviluppo dei lavori.

Un ringraziamento che parte dal profondo del mio cuore desidero rivolgerlo al Prof. Grigolini. Paolo mi ha dato l'onore e la possibilità di collaborare con il suo gruppo di ricerca presso la University of North Texas di Denton, regalandomi tra l'altro i sette mesi più belli e di più intensa crescita della mia vita. La Sua autorevolezza scientifica, ma allo stesso tempo la Sua estrema umiltà e generosità, fanno di Lui un maestro di vita, che sarà per me sempre modello di ispirazione per il mio percorso professionale ed esistenziale.

Un'ultima riconoscenza nell'ambito accademico voglio sentitamente porla al Prof. Mauro Dell'Orco, che ha condiviso con me i viaggi casa-lavoro, fornendomi giornalmente preziosi consigli e pareri in ambito lavorativo, nonché allietando con la simpatia che lo contraddistingue la quotidianità di questi tre anni.

Rivolgo la mia più viva gratitudine alle radici salde e profonde della mia persona: i miei genitori. Il loro esempio di vita, i loro sacrifici, i valori con i quali sono cresciuto, i loro incoraggiamenti e le loro premure sono state indispensabili per permettere la realizzazione delle mie aspirazioni. Grazie ai nonni che non ci sono più, in particolare a mio nonno Nicola, luce del mio cammino e matrice dell'amore trasmessomi per le scienze.

Un grazie di cuore va ai miei colleghi e compagni di viaggio, Antonio ed Elia; tra noi c'è sempre stata una collaborazione eccezionale, ci siamo dati forza l'un l'altro sin dall'inizio, dando ininterrottamente il massimo, in un'equa, sana e proficua competizione.

Grazie infine ai miei amori e agli amici di sempre, a tutti coloro che mi son stati accanto in questi anni. La vostra dolce presenza, ciò che siete capaci di donarmi, tutto quello che abbiamo condiviso e vissuto insieme, costituiscono un tesoro enorme che custodirò sempre gelosamente nei miei ricordi.

Infine, l'ultimo pensiero va a me, alla quotidiana determinazione, tenacia e perseveranza, che ho esibito in questa sfida, che mi ha portato qualche tribolazione, qualche gioia, tante soddisfazioni ed una rinnovata consapevolezza di me, dei miei limiti e delle mie abilità.

DENTON, DECEMBER 2016 I. DE VINCENZO

Contents

1 INTRODUCTION 1

2 COLLECTIVE INTELLIGENCE IN NATURAL AND ARTIFICIAL SYSTEMS 5

2.1 COLLECTIVE INTELLIGENCE IN ANIMALS 5

2.1.1 The mechanisms of collective behaviors 7

2.1.2 Principles and properties of self-organizing processes 9

2.1.3 Categorizing the collective behaviors of social insects 10

2.2 COLLECTIVE INTELLIGENCE IN HUMANS 12

2.2.1 Possibilities and limitations of Swarm Intelligence in human groups 12

2.2.2 Consequences of Swarm Intelligence research on human groups 14

2.3 COLLECTIVE ARTIFICIAL INTELLIGENCE: SWARM ROBOTICS 15

2.3.1 Tasks in Swarm Robotics 17

2.3.2 Towards Real World Application 20

3 BIOLOGICALLY INSPIRED OPTIMIZATION METHODS 23

3.1 GENETIC ALGORITHM 26

3.2 ANT COLONY OPTIMIZATION 29

3.3 PARTICLE SWARM OPTIMIZATION 32

3.4 DIFFERENTIAL EVOLUTION 35

3.5 ARTIFICIAL BEE COLONY 37

3.1 GLOWWORM SWARM OPTIMIZATION 39

4 DECISION MAKING MODEL OF HUMAN GROUPS 43

4.1 THE MODEL 44

4.2 MEASURING THE PERFORMANCE OF THE DECISION-MAKING PROCESS 49

4.3 SIMULATIONS AND RESULTS 50

4.4 CRITICAL TRANSITION AND COLLECTIVE INTELLIGENCE 55

4.5 CONCLUSIONS 61

5 THE EFFECT OF SOCIAL NETWORK STRUCTURE ON TEAM PERFORMANCE 63

5.1 THE COMPUTATIONAL MODEL ADAPTED TO THE CASE OF STUDY 64
5.2 SIMULATION AND RESULTS 70
5.3 CONCLUSIONS 77

6 HUMAN GROUP OPTIMIZATION ALGORITHM 79

6.1 INTRODUCTION 80
6.2 THE HUMAN GROUP OPTIMIZATION ALGORITHM 83
6.3 SIMULATION AND RESULTS 84
6.4 COMPARISON WITH OTHER OPTIMIZATION ALGORITHMS 89
6.5 CONCLUSIONS 91

7 CONCLUSIONS 93

8 APPENDIX 97

8.1 THE NK FITNESS LANDSCAPE 97
8.2 THE GLAUBER DYNAMICS OF ISING MODEL ON GENERAL GRAPHS 102
8.3 DETAILED BALANCE CONDITION 105
8.4 THE EXPONENTIAL DISTRIBUTION OF EVENTS AND THE GILLESPIE ALGORITHM 106
8.5 SIMULATED ANNEALING AND OPTIMAL INITIAL TEMPERATURE 110
8.6 NOTES OF INFORMATION THEORY 116
8.7 THE RESCALED NK LANDSCAPE 121

9 REFERENCES 123

1 Introduction

"We cannot solve our problems
with the same thinking we used
when we created them."
Albert Einstein

The ability of animal and human groups in solving complex problems is widely recognized in nature. Flocks of birds, schools of fish, colonies of ants and bees, are just a few of the best-known examples in the animal world, where interaction and collaboration between agents with limited ability and capabilities, allow them to solve problems exceeding individual skills (foraging, defending from predators, building structures, etc.). This phenomenon is known as Swarm Intelligence (SI) (Beni, 1989).

Similarly, human groups, exploiting the potentialities of social interactions and sharing their limited knowledge, are collectively able to achieve much better performance than single individuals can do. This specific ability of human groups has been defined as Collective Intelligence (Woolley, Chabris, Pentland, Hashmi, & Malone, 2010).

Surprisingly, the complexity of these cooperative behaviors and structures does not reflect at all the relative simplicity of the individual actions. A single insect, for example, is not able to assess the global situation of its colony, to centralize information about the state of other insects, and to find by itself an efficient solution, while the society to which it belongs, finds, as a whole, a solution very easily (Camazine, et al., 2001).

The superior ability of groups in solving tasks originates from *collective decision making*: agents (animals, robots, humans) make choices, pursuing their individual goals (forage, survive, etc.) based on their own knowledge and amount of information (position, sight, etc.), and adapting their behavior to the actions of the other agents. Even though the single agents possess a limited knowledge, and the actions they perform are usually very simple, the collective behavior, enabled by group-leaving and social interactions, allows the knowledge and information sharing, leading to the development of a superior intelligence of the group (Clément, 2013), (Couzin, 2005), (Sumpter et al., 2009), (Ward et al., 2008), (Arganda et al., 2012), (Ward et al., 2011), (Perez-Escudero, 2011), (Watts, 2002), (Turalaska et al., 2009), (Wang & Szolnoki, 20013).

This dissertation deals with the emergence and exploitation of the collective intelligence of human groups. Differently from the SI, the topic of the Collective Intelligence is relatively new in the field of complexity science and just a few studies have been conducted on it.

The concept of collective intelligence is generally associated with groups of agents, cognitively different, but overall superior in terms of rational capabilities, respect to the extremely simple animals to which we refer talking about Swarm Intelligence. This principally means that they have an albeit limited overview, taking decisions trying to maximize an objective function, and not simply looking for the consensus, that anyway is the fundamental tool to get the superior intelligence of the group.

Different studies have been proposed in the literature, trying to model the human behavior (Bordogna & Albano, 2007), (Turalaska et al., 2009), (Turalaska, West & Grigolini, 2009). All of them recognize the *imitation* as the fundamental mechanism by which the phenomena of crowds, fads, and crime could be understood. Imitation is, in fact, the basic of learning: children imitate their parents and peers, entrants into an industry imitate more established firms, governments in less developed countries imitate governments in more developed countries. However, if everyone strictly imitates, improvements in choice cannot occur, while it is widely recognized that humans' cooperation led to achieving astonishing performances in the history of humanity. One of the main contributions of this dissertation has been to fill this gap, trying to model, consensus seeking apart, the rational evaluation that characterizes human beings, to compare alternative strategies in terms of costs and benefits and efficiently solve a problem.

This work is organized as follow. Chapter 2 aims to review the mechanisms beyond the swarming behaviors in natural systems, focusing on their properties, potentialities, and limitations, as well as providing the state of the art in the developing field of swarm robotics.

In chapter 3, some of the most known biologically inspired optimization algorithms, are introduced, highlighting their variants, merits and drawbacks.

In chapter 4, the author introduces a new decision-making model (DMM), firstly proposed by Carbone and Giannoccaro (Carbone & Giannoccaro, 2015) for solving complex combinatorial problems, showing a wider detailed analysis of its features and potentialities. The DMM attempts to capture the previously mentioned two drivers of the humans' behaviors in groups, i.e., *self-interest* and *consensus seeking*: individuals make choices based on rational calculation and self-interested motivations. Agent's choices are made by optimizing the perceived fitness value, which is an estimation of the real one, based on the

level of agent's knowledge (Conradt, 2003), (Turalska & West, 2014), (Conradt, 2012). However, any decision made by an individual is influenced by the relationships he/she has with the other group members. This social influence pushes the individual to modify the choice he/she made, for the natural tendency of humans to seek consensus and avoid conflict with people they interact with (Di Maggio & Powell, 1983). Consequently, effective group decisions spontaneously emerge as the result of the choices of multiple interacting individuals.

We found that a moderate strength of social interactions allows for knowledge transfer among the members, leading to higher knowledge level of the group as a whole. This mechanism, coupled with the ability to explore the fitness landscape, strongly improves the performance of the decision-making process. As in the case of a collection of spins, the dynamics of such a system is characterized by a phase transition from low to high values of the consensus (magnetization), associated with the emergence of a collective superior intelligence of the population. We identified that the threshold value of the social interaction strength, at which the entire group is characterized by a higher degree of collective intelligence, is just the critical threshold at which the flow of information from the fitness landscape to the group of agents is maximized, thus improving the abilities of the group to explore the fitness landscape searching for the optimal solution.

In Chapter 5 an application of the DMM to the simulation of a management problem is reported. We simulate how a team of individuals with different expertise, appointed to design a new product, converge towards a shared solution of the design process by interacting with each other. We suppose the interactions among individuals take place for hierarchical or social reasons, investigating respectively the influence of the team hierarchical structure and social network topology on the team performances, and providing also indications about how to effectively design a team. Note that a similar formulation can be easily adapted to the simulation of any kind of social decision-making problems, i.e. modeling of parliaments and squads' dynamics, coordination of human crowds, etc.

In chapter 6 the author introduces a novel optimization algorithm belonging to the class of swarm intelligence optimization methods. The proposed algorithm, referred as Human Group Optimization (HGO), is developed within the previously mentioned DMM (Carbone & Giannoccaro, 2015) and emulates the collective decision-making process of human groups. To this end, a continuous-time Markov process is proposed to describe the behavior of a population of socially interacting agents, modeling how humans in a group modify their

opinions driven by self-interest and consensus seeking. By tuning some control parameters, it is possible to make the system undergo a critical transition towards a state of high consensus which is always accompanied by an analogs transition from low to high group fitness values, leading to the emergence of the collective intelligence of the group. While this state being active, a cooling schedule is applied to make agents closer and closer to the optimal solution, while performing their random walk on the fitness landscape. To test the ability of the HGO algorithm, we compare its performance with those of the Simulated Annealing (SA), and Genetic Algorithm (GA) in solving *NP*-complete problems, consisting in finding the optimum on a fitness landscape, the latter generated within the Kauffman *NK* model of complexity (Kauffman & Levin, 1987), (Kauffman & Weinberger, 1989).

Last but not least, Chapter 8 contains all the mathematical tools and the basic notions, necessary for a complete understanding of the models and procedures mentioned in the work.

2 Collective Intelligence in Natural and Artificial Systems

"The whole is greater than the sum of its parts"
Aristotele

Complexity science has shown that collective behaviors in animal and human groups allow them to solve problems that go beyond the capacity of each single agent; this phenomenon is known as **Swarm Intelligence** (SI) (Beni, 1989).
Swarming systems, be them natural or artificial, are all characterized by similar features: (i) *lack of central controller* overseeing the collective dynamics, that instead emerge through self-organization, (ii) *local perception of the environment* leading to a certain level of global knowledge by means of effective distributed information sharing, and (iii) a *high degree of adaptation* to rapidly changing circumstances.

Not all kinds of collective behavior are evidence of SI. A flock of birds or a crowd of humans in which individuals simply stay together through social attraction are not necessarily examples of SI. Nevertheless, grouping is known to be advantageous for many reasons other than increased cognitive abilities, so it is likely that whenever individuals interact and live in groups, there is a potential for a SI developing.

This chapter aims at reviewing the mechanisms beyond a swarming behavior, taking in consideration animals, humans and artificial systems.

2.1 Collective Intelligence in Animals

Swarm intelligence, as a scientific discipline, was born from biological insights about the incredible abilities of social insects to solve their everyday-life problems (Bonabeau, Dorigo, & Theraulaz, 1999). Their colonies display attractive behaviors that combine efficiency with both flexibility and robustness (Camazine, et al., 2001). From the traffic management on a foraging network (Burd, 2006), (Couzin & Franks, 2003), (Dussutour, Deneubourg, & Fourcassié, 2005), (Vittori, et al., 2006), to the building of efficient structures (Buhl, Deneubourg, Grimal, & Theraulaz, 2005); (Buhl, et al., 2006), (Theraulaz, Gautrais, Camazine, & Deneubourg, 2003), (Tschinkel W. R., 2003), (Tschinkel W. R., 2004), along with the dynamic task allocation between workers (Beshers & Fewell, 2001), (Bonabeau,

Theraulaz, & Deneubourg, 1998), (Deneubourg, Goss, Pasteels, Fresneau, & Lachaud, 1987), (Gordon, 1996), examples of complex and sophisticated behaviors are numerous and diverse among animals (Bonabeau E. , Theraulaz, Deneubourg, Aron, & Camazine, 1997), (Camazine, et al., 2001), (Detrain & Deneubourg, 2006).

Surprisingly, the complexity of these collective behaviors and structures does not reflect at all the relative simplicity of the individual behaviors. Taking in consideration insects for example, we can say that in most cases, a single insect is not able to find by itself an efficient solution to a colony problem, while the society to which it belongs finds *as a whole* a solution very easily (Camazine, et al., 2001). Behind this *organization without an organizer* there are several hidden mechanisms which enable animal societies to cope with uncertain situations and to find solutions to complex problems.

Insects apart, also other biological systems [Fig. 2.1] share collective properties such as colonies of bacteria or amoeba (Ben-Jacob, et al., 1994), (Ben-Jacob, Cohen, & Levine, 2000), fish schools (Grünbaum, Viscido, & Parrish, 2005); (Parrish, Viscido, & Grünbaum, 2002), bird flocks (Reynolds, 1987), sheep herds (Gautrais, Michelena, Sibbald, Bon, & Deneubourg, 2007) or even crowds of human beings (Helbing, Molnàr, Farkas, & Bolay, 2001). Nevertheless, we will focus on the cooperative behaviors of social insects because they represent the largest research corpus from a theoretical and experimental point of view and because their underlying principles are very close to those found in others animals.

Fig. 2.1, Biological system sharing collective properties. (a) fish school of Blue jack mackerel merge into a torus that confuses predators; (b) Birds coordinate their speed and direction with just a half dozen of their closest murmuration-mates. These interactions are enough to steer the entire group in the same direction; (c) Sheep dogs control the flocking behavior ; (d) crowds of human beings.

2.1.1 The mechanisms of collective behaviors

For a long time, the collective behavior of social insects has remained a fascinating issue for naturalists. Individual insects were assumed to possess something like a representation of the task to solve and then they were supposed to use it to make appropriate decisions (Thorpe, 1963).

Nevertheless, most of the works that have been done in the last 40 years (Theraulaz G. B., 1998) revealed that individual insects do not need any representation or explicit knowledge of the global problem to solve. A single insect is not able to assess a global situation, to centralize information about the state of its entire colony and then to control the tasks to be done by the other workers. There is no supervisor in these colonies.

The first serious theoretical explanation to the organization of social insects' activities was provided by French biologist Pierre-Paul Grassé, who introduced the concept of **stigmergy** to explain *building activity in termites* (Grassé, 1959). Grassé showed that information coming from the local environment and the work in progress can guide individual activity. For instance, each time a worker performs a building action, the shape of the local configuration that triggered this action is changed. The new configuration will then influence other specific actions from the worker or potentially from any other workers in the colony. This process leads to an almost perfect coordination of the collective work and may give us the impression that the colony is following a well-defined plan.

A good example of the stigmergic behavior is provided by nest building in social wasps. Building activities are driven by the local configuration of cells detected by the wasps on the nest (Karsai & Theraulaz, 1995). To decide where to build a new cell, wasps use the information provided by the local arrangement of cells on the outer circumference of the comb. They have a greater probability of adding new cells to a corner area where three adjacent walls are already present, while the probability of starting a new row, by adding a cell on the side of an existing row, is very low (Camazine, et al., 2001).

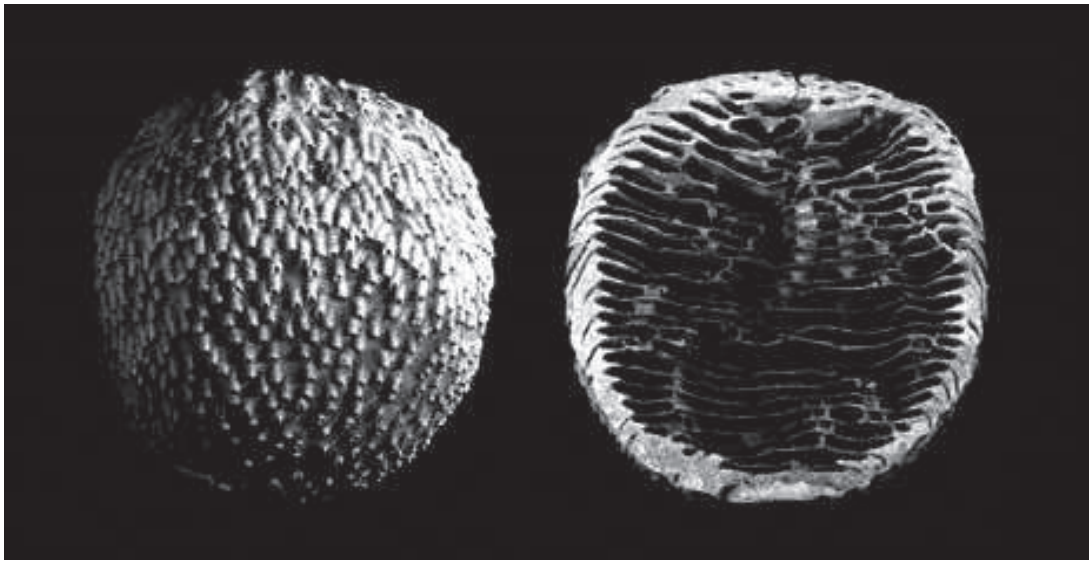


Fig. 2.2, An external view and a cross section of an Apicotermes lamani nest resulting from the coordination of workers building activities.

Another example of a stigmergic behavior is food recruitment in ants (Hölldobler & Wilson, 1990). Ants communicate with each other using pheromones. These pheromones are chemical substances that attract other ants. For instance, once an ant has found a food source, she quickly comes back to the nest and lays down a pheromone trail. This trail will then guide other workers from the nest toward the food source. When the recruited ants come back to the nest, they lay down their own pheromone on the trail and reinforce the pathway. The trail formation, therefore, results from a positive feedback: the more ants use a trail, the more attractive the trail becomes. Of course, the trail will disappear after some time if the reinforcement is too slow, which may occur when the food source becomes exhausted. This trail recruitment system is not only a mechanism used to quickly assemble many foragers around a food source, it also enables a colony to make efficient decisions such as the selection of the shortest path leading to a food source.

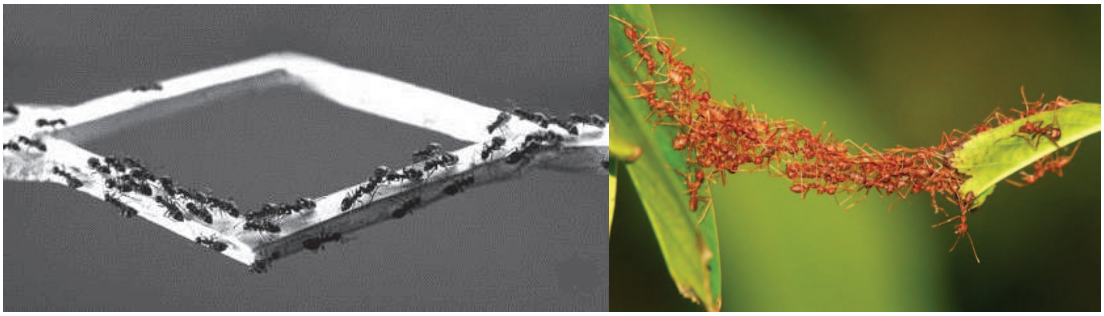


Fig. 2.3, Ant colonies. (a) Collective selection of one foraging path over a diamond-shaped bridge leading to a food source by workers in the ant Lasius niger. (b) Weaver ant workers cooperate to form chains of their own bodies, allowing them to cross wide gaps and pull leaves together.

Today, we know that most collective decisions in social insects arise through the competition among different types of information that can be amplified in various ways. In the case of path selection by ants, the information is conveyed by the pheromone trail. However, environmental constraints, such as the distance between the nest and the food source, affect this positive feedback. In particular, any constraint that modulates the rate of recruitment or the trail concentration on a branch can lead that branch to lose, or win, its competition against the other one. Thus, an efficient decision can be made without any modulation of individual behavior and without any sophisticated cognitive processing at the individual level.

2.1.2 Principles and properties of self-organizing processes

Self-organization is the major component of a wide range of collective behaviors in social insects, from the thermoregulation of bee swarms to the construction of nests in ants (Bonabeau E. , Theraulaz, Deneubourg, Aron, & Camazine, 1997), (Camazine, et al., 2001). Self-organization is a set of dynamical mechanisms whereby structures appear at the global level of a system from interactions among its lower-level components, without being explicitly coded at the individual level. It relies on four basic ingredients:

1. A **positive feedback** that results from the execution of simple behavioral "rules of thumb" that promote the creation of structures. Trail recruitment to a food source is a kind of positive feedback which creates the conditions for the emergence of a trail network at the global level.
2. A **negative feedback** that counterbalances positive feedback and that leads to the stabilization of the collective pattern. In the example of ant foraging, it may result from the limited number of available foragers, the food source exhaustion, and the evaporation of pheromone or a competition between paths to attract foragers.
3. The **amplification of fluctuations** by positive feedbacks. Social insects are well known to perform actions that can be described as stochastic. Such random fluctuations are the seeds from which structures nucleate and grow. Randomness is often crucial because it enables the colony to discover new solutions.
4. **Multiple direct or stigmergic interactions** among individuals to produce apparently deterministic outcomes.

Moreover, self-organization is also characterized by a few key properties:

1. Self-organized systems are **dynamic**. As stated before, the production of structures as well as their persistence requires permanent interactions between the members of the colony and with their environment. These interactions promote the positive

feedbacks that create the collective structures and act for their subsistence against negative feedbacks that tend to eliminate them.

2. Self-organized systems exhibit **emergent properties**. They display properties that are more complex than the simple contribution of each agent. These properties arise from the nonlinear combination of the interactions between the members of the colony.
3. Nonlinear interactions lead self-organized systems to **bifurcations**. A bifurcation is the appearance of new stable solutions when some of the system's parameters change. This corresponds to a qualitative change in the collective behavior.
4. Self-organized systems can be **multi-stable**. Multi-stability means that, for a given set of parameters, the system can reach different stable states depending on the initial conditions and on the random fluctuations.

2.1.3 Categorizing the collective behaviors of social insects

According to (Garnier, Gautrais, & Theraulaz, 2007) four functions in the collective behaviors of insects are recognized: *coordination*, *cooperation*, *deliberation* and *collaboration*.

1. **Coordination** is the appropriate organization in space and time of the tasks required to solve a specific problem. This function leads to specific spatial-temporal distributions of individuals, of their activities and/or of the results of their activities in order to reach a given goal. For instance, coordination occurs in the organization of the displacement in bee and locust swarm (Buhl, et al., 2006); (Janson, Middendorf, & Beekman, 2005), or in the exploitation of food sources by pheromone trail laying ants (Hölldobler & Wilson, 1990), (Traniello & Robson, 1995).
2. **Cooperation** occurs when individuals achieve together a task that could not be done by a single one. The individuals must combine their efforts to successfully solve a problem that goes beyond their individual abilities. Cooperation is obvious in large prey retrieval when a single individual is too weak to move a food item. Many cases of cooperative transport of prey were reported for several ant species such as weaver ants Oecophylla longinoda, army ants Eciton burchelli (Franks, 1986).
3. **Deliberation** refers to mechanisms that occur when a colony faces several opportunities. These mechanisms result in a collective choice for at least one of the opportunities. For instance, when ants of the species Lasius niger have discovered several food sources with different qualities or richness, or several paths that lead to

a food source, they generally select only one of the different opportunities. In this case, the deliberation is driven by the competition between the chemical trails, leading ants to forage at the richer food source and to travel along the shorter path toward the food source (Beckers, Deneubourg, Goss, & Pasteels, 1990), (Goss, Aron, Deneubourg, & Pasteels, 1989).

4. **Collaboration** means that different activities are performed simultaneously by groups of specialized individuals, for instance foraging for prey or tending brood inside the nest (Gordon, 1996), (Wilson, 1971). This specialization can rely on a pure behavioral differentiation as well as on a morphological one and be influenced by the age of the individuals. The most conspicuous expression of such division of labor is the existence of castes. For instance, in leaf-cutter ants, workers may belong to four different castes and their size is closely linked to the tasks they are performing (Hölldobler & Wilson, 1990). Only the workers whose head size is larger than 1.6 millimeters are able to cut the leaves that are used to grow a mushroom that is the main food source of these colonies. On the contrary, only the tiny workers whose head size is about 0.5 millimeters are able to take charge of the cultivation of the mushroom.

As exemplified in this subsection, the organization of collective behaviors in social insects can be understood as the combination of the four functions. Together, the four functions of organization produce solutions to the colony problems and may give the impression that the colony as a whole plans its work to achieve its objectives.

2.2 Collective Intelligence in Humans

The underlying perception in most biological (non-human) case studies of SI has been that the individual animal is cognitively relatively simple and restricted in what it can achieve, whereas the group collectively is capable of astonishing feats. Differently, in humans, there is not only interindividual variation in cognitive abilities, but there are also some individuals that are high performers by any standard (Kerr & Tindale, 2004).

Thus, emphasis in psychology and management has focused on assessing if a group can outperform high-performing individuals, in trying to find the limits of what a group of a given size and composition can collectively achieve and in understand how to create collaborative organizations that result more productive and intelligent.

Collective intelligence in humans was defined by Thomas Melone (Malone & Bernstein, 2015), like a condition in which *group of individuals acting collectively in a way that seems intelligence.* By this definition, collective intelligence has existed for a very long time: army, countries, families, are all examples of groups of people working together, in a way that "sometimes" seems intelligent.

Experimental tests brought to the deduction that there are two main factors determining the collective intelligence of a group: how people are involved in the group and how they work together. They indicate that SI performs well if there is diversity and independence of opinions, an incentive for truthful reporting, and if estimates of individuals are only hampered by imprecision and not by a systematic bias (Kerr & Tindale, 2004), (Wolfers & Zitzewitz, 2004), (Laughlin, 2003).

2.2.1 Possibilities and limitations of Swarm Intelligence in human groups

Media articles often suggest that SI could be the answer to every decision-making or forecasting problem in modern society. The difference between areas in which SI can and cannot contribute is, however, easily illustrated by a simple example. At a biomimetic exhibition in Berlin, Germany, researchers (Krause, Ruxton, & Krause, 2010) presented the public two tasks. In the first problem, they needed to estimate the number of marbles in a large glass jar. For the second problem, they had to estimate how many times a coin needs to be tossed for the probability that the coin shows heads (and not tails) on all occasions to be roughly as small as that of winning the German lotto, which is a probability of 1 in 14 million. For the first problem, the collective estimate came within 1.5% of the real value [Fig. 2.4], an impressive performance of SI, while in the second case the collective guess

was poor. For a person with a background in combinatorics, this second task involves only a quick calculation that always arrives at the correct answer of 24 coin tosses. Clearly, expert knowledge would be superior here.

An interesting difference is so between *imprecision* and *bias* in this context. For the marbles, the individual estimate is imprecise and uncorrelated guesses can result in a close approximation of the real value. In the second case, there is a huge systematic bias preventing useful information extraction. Most real-life problems will have components of both imprecision and bias, and the general rule would be that the greater the imprecision component (relative to the bias component), the greater the potential for SI solutions.

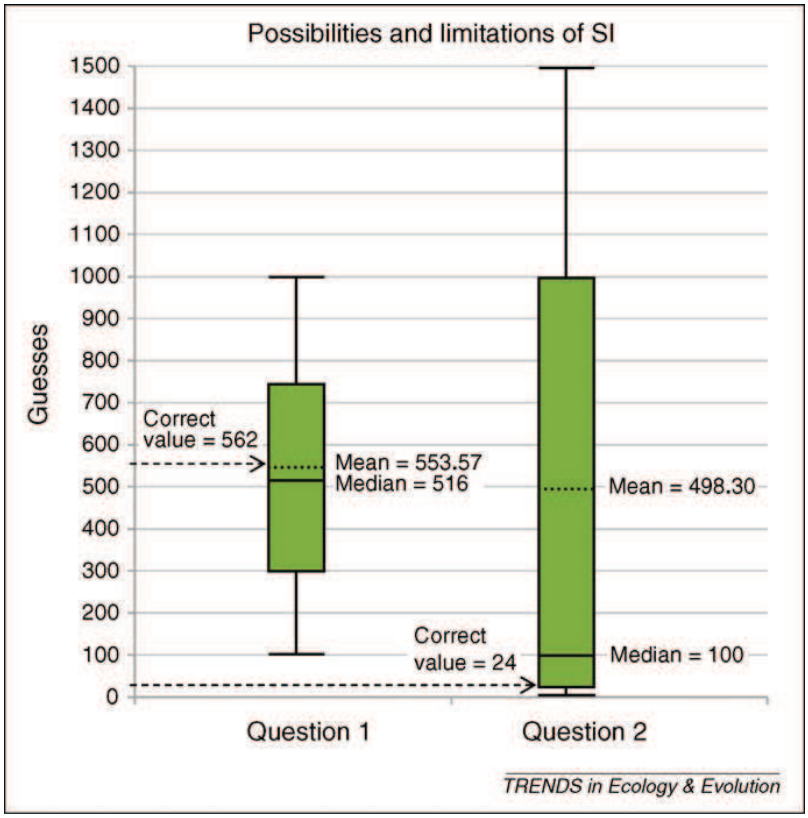


Fig. 2.4, Possibilities and limitations of SI (Krause, Ruxton, & Krause, 2010). Members of the public were presented with two questions. The first question was to estimate the number of marbles in a glass jar and the second question was to estimate the number of times a coin needs to be tossed for the probability that the coin shows heads on all occasions to be roughly as small as that of winning the German lotto, which is a probability of 1 in 14 million. Box-and-whisker plots are shown for the questions 1 (number of marbles) and 2 (number of coin tosses). The boxes are bounded by the 0.25 and 0.75 quantiles, and the whiskers end at the 0.1 and 0.9 quantiles. The means are marked by dotted lines and the correct values by dashed arrows. For question 1 (N = 2057), the absolute deviations of mean and median from the correct value are small (1.5% for the mean, and 8.2% for the median). For question 2 (N = 1953), however, the mean is a poor approximation to the correct value (absolute deviation = 1976%). The median is better than the mean, but is still far away from the correct value (absolute deviation = 317%).

2.2.2 Consequences of Swarm Intelligence research on human groups

The finding that the judgment of a diverse group can outperform an expert or even a small group of experts under certain circumstances (Laughlin, 2003) has led to speculation that SI developments could make experts obsolete to the extent that even company CEOs might be in less demand in the future (Surowiecki, 2004). Although this seems unlikely, probably a shift might be seen in the type of experts that are needed, towards experts who know the mechanisms to harness and implement SI.

In social insects, the individuals might collectively be able to solve cognitive problems. However, even when they have arrived at a solution, a single ant or bee is never going to be in possession of the overall information (or solution). By contrast, humans can purposefully set out to use SI principles to their benefit to gain, for instance, a competitive advantage in business by better predicting market developments.

The point is that the whole SI mechanism (data collection, processing and solution) can be used by single experts or expert teams. Therefore, the potential user of SI in animals and in humans is fundamentally different in this respect. In animals, SI acts as an enabler for a group of often highly interdependent individuals; in humans, it can be an enabler as well as a tool that can be used to aid decision making.

We may assist to the shift from traditional hierarchies to flatten organizational structures. For years, pockets of U.S. military have been slowly taking decisions out of the hands of high-ranking commanders and entrusting them to teams of soldiers, who are told what problems to solve but not how to solve them. Probably nature is teaching us that sometimes the best way to gain power is to give it away. Linus Torvalds, the developer of the Linux open-source operating system, or Pierre Omidyar, the founder of Ebay, for example, gave power away, respectively to thousands of programmers and customers all over the world, and they were rewarded with a different kind of power.

2.3 Collective Artificial Intelligence: Swarm Robotics

Swarm robotics is the study of how large number of simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among agents and between the agents and the environment, (Şahin, 2005).

It takes its inspiration from societies of insects that can perform tasks that are beyond the capabilities of the individuals. In order to be able to differentiate this definition from other multi-robot types of systems, it is complemented with a set of criteria:

1. The robots of the swarm must be **autonomous**, able to sense and actuate in a real environment.
2. The number of robots in the swarm must be **large** or at least the control rules allow it.
3. Robots must be **homogeneous**. There can exist different types of robots in the swarm, but these groups must not be too many.
4. The robots must be **incapable** or inefficient respect to the main task they have to solve, i.e. they need to collaborate in order to succeed or to improve the performance.
5. Robots have only **local communication** and sensing capabilities. It ensures the coordination is distributed, so scalability becomes one of the properties of the system.

Iocchi et al. present a taxonomy (Iocchi, Nardi, & Salerno, 2001) structured in different levels. The first level is *Cooperation*, which includes a situation in which several robots perform a common task. The second level is *Knowledge*, which distinguishes whether robots know of the existence of other robots (*Aware*) or not (*Unaware*). The third level is *Coordination*, to differentiate the degree in which robots take into account the actions executed by other robots. According to the authors of the taxonomy, this can be *Strongly Coordinated*, *Weakly Coordinated*, or *Not Coordinated*. The last level is *Organization*, which distinguishes between *Centralized* systems, where there exists a robot that is in charge of organizing other robots' work, and *Distributed* systems, where robots are autonomous in their decisions, i.e there are no leaders. According to this taxonomy, swarm robotic systems are *Cooperative*, *Aware*, *Strongly Coordinated* or *Weakly Coordinated*, and *Distributed*.

A schematic of the taxonomy is shown in Fig. 2.4, where for each level the corresponding type of system is marked in dark grey for a swarm-robotic system.

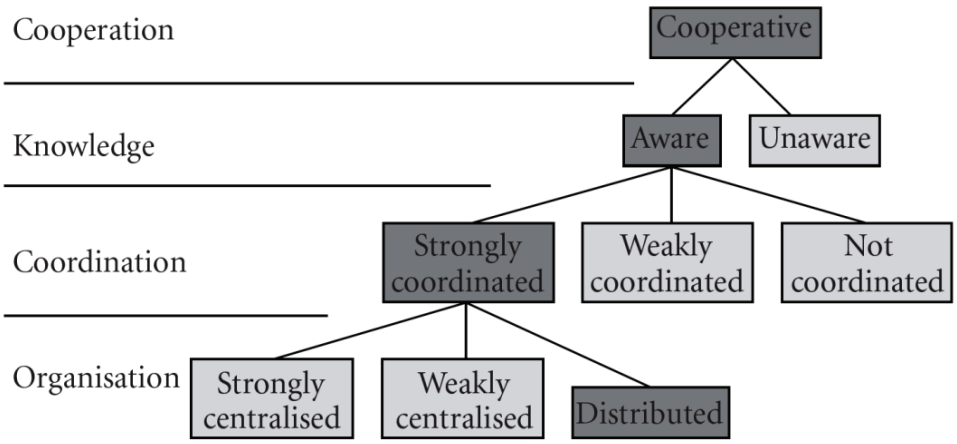


Fig 2.5, Taxonomy from (Iocchi, Nardi, & Salerno, 2001). For each level the corresponding type of system is marked in dark grey for a swarm robotic system.

According to (Ronald Arkin, 1998), advantages of multi-robotic approaches, compared to single-robot systems, are the following:

1. **Improved performance**: if tasks can be decomposable then by using parallelism, groups can make tasks to be performed more efficiently.
2. **Task enablement**: groups of robots can do certain tasks that are impossible for a single robot.
3. **Distributed sensing**: the range of sensing of a group of robots is wider than the range of a single robot.
4. **Distributed action**: a group of robots can operate in different places at the same time.
5. **Fault tolerance**: under certain conditions, the failure of a single robot within a group does not imply that the given task cannot be accomplished, thanks to the redundancy of the system.

Drawbacks are the following.

1. **Interference**: robots in a group can interfere between them, due to collisions, occlusions, and so forth.
2. **Uncertainty concerning other robots 'intentions**: coordination requires to know what other robots are doing. If this is not clear, robots can compete instead of cooperate.
3. **Overall system cost**: the fact of using more than one robot can make the cost bigger. This is ideally not the case of swarm-robotic systems, which intend to use many cheap and simple robots which total cost is under the cost of a more complex single robot carrying out the same task.

2.3.1 Tasks in Swarm Robotics

A collection of the most representative tasks assigned to swarming robots is depicted in this section. They are presented in increasing order of complexity.

1. **Aggregation**. The goal of aggregation is to group all the robots of a swarm in a region of the environment. Despite being a simple collective behavior, aggregation is a very useful building block, as it allows a swarm of robots to get sufficiently close one another so that they can interact. Aggregation is a very common behavior in nature. For example, it can be observed in bacteria, cockroaches, bees, fish and penguins (Camazine, et al., 2001). Garnier et al. (2005) developed a system in which robots are used to replicate the behavior observed in cockroaches by (Janson, Middendorf, & Beekman, 2005). The robots can collectively aggregate in a circular arena. Other examples of an aggregation behavior based were developed by (Soysal & Sahin, 2005) and (Trianni, Groß, Labella, Şahin, & Dorigo, 2003) [Fig. 2.5].

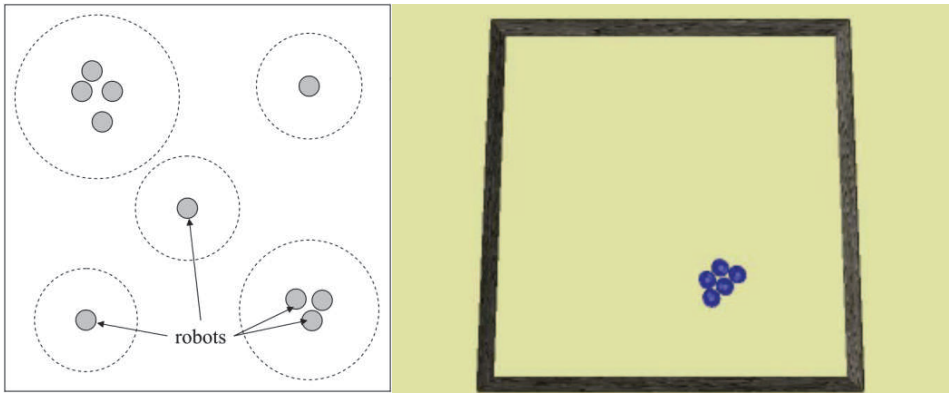


Fig. 2.6, Examples of the aggregation collective behavior. (a) (Soysal & Sahin, 2005) (b) (Trianni, Groß, Labella, Şahin, & Dorigo, 2003).

2. **Dispersion**. The aim of dispersion is to distribute the robots in space to cover it as much area as possible, usually without losing the connectivity between them.
3. **Patter Formation**. Pattern formation aims at deploying robots in a regular and repetitive manner. Robots usually need to keep specific distances between each other to create the desired pattern. Pattern formation can be found both in biology and in physics. Some biological examples are the spatial disposition of bacterial colonies and the chromatic patterns on some animal's fur (Meinhardt & Meinhardt, 1982). Some physics examples are molecules distribution and crystal formation (Langer, 1980), and Benard cells (Getling, 1998).
4. **Collective Movement**. Collective movement is the problem of how to coordinate a group of robots and move them together as a group in a cohesive way. It can also

serve as a basic behavior for more complicate tasks. It can be classified into two types: formations and flocking. In the former, robots must maintain predetermined positions and orientations among them. On the other hand, in flocking, robots' relative positions are not strictly enforced (W. M. Spears, 2004).

5. **Task Allocation**. Task allocation is a collective behavior in which robots distribute themselves over different tasks. The goal is to maximize the performance of the system by letting the robots dynamically choose which task to perform. Task allocation can be observed in natural systems such as ant and bee colonies, (Theraulaz G. B., 1998). For example, in ant or bee colonies, part of the swarm can perform foraging while another part looks after the larvae. Task allocation is not fixed but can change over time. Jones and Matarić (Jones & Matarić, 2003) present a distributed and scalable algorithm for labor division in swarms of robots. Each robot maintains a history of the activities performed by other robots based on observation and independently performs a division of labor using this history.
6. **Source Search**. Swarm robotics can be very useful in search tasks, especially those in which the spatial pattern of the source can be complex as in the case of sound or odor.
7. **Collective Transport of Objects**. Swarm robotics is very promising in solving the problem object transportation. The use of many robots can represent an advantage because of cooperation handling one object. In addition, the possible parallelism dealing with different objects by several robots at the same time might improve the performance.

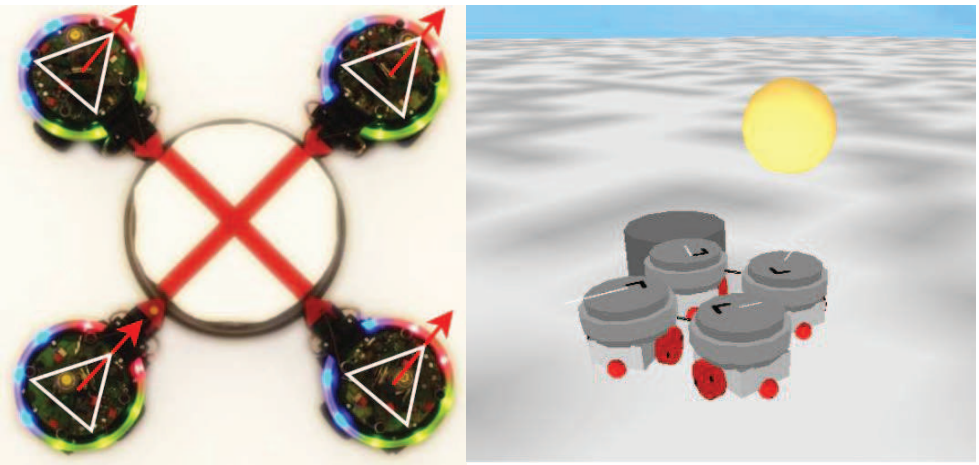


Fig 2.7, Examples of the collective transport behavior. (a) From (Campo, Nouyan, Birattari, Groß, & Dorigo, 2006) (b) From (Baldassarre, Parisi, & Nolfi, 2006).

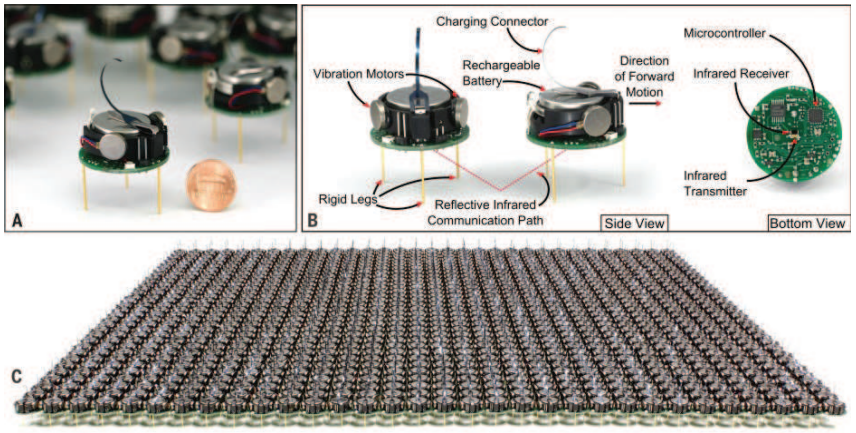


Fig. 2.8, Kilobot swarm robot (Rubenstein, Cornejo, & Nagpal, 2014). (A) A Kilobot robot, shown alongside a U.S. penny for scale. (B) Each Kilobot has an onboard microcontroller for executing programs autonomously, two vibration motors for moving straight or turning on a flat surface, and a downward-facing infrared transmitter and receiver. Robots communicate with others within a range of 10 cm (roughly three robot diameters) by reflecting infrared light off the table below. Communicating robots can evaluate relative distance by measuring the strength of the received infrared signal, but they cannot sense relative bearing (angle). (C) A 2^{10} Kilobot swarm. The Kilobot design allows for all operations on the entire swarm (charging, programming, etc.) to take a constant time to complete, independent of the number of robots in the swarm.

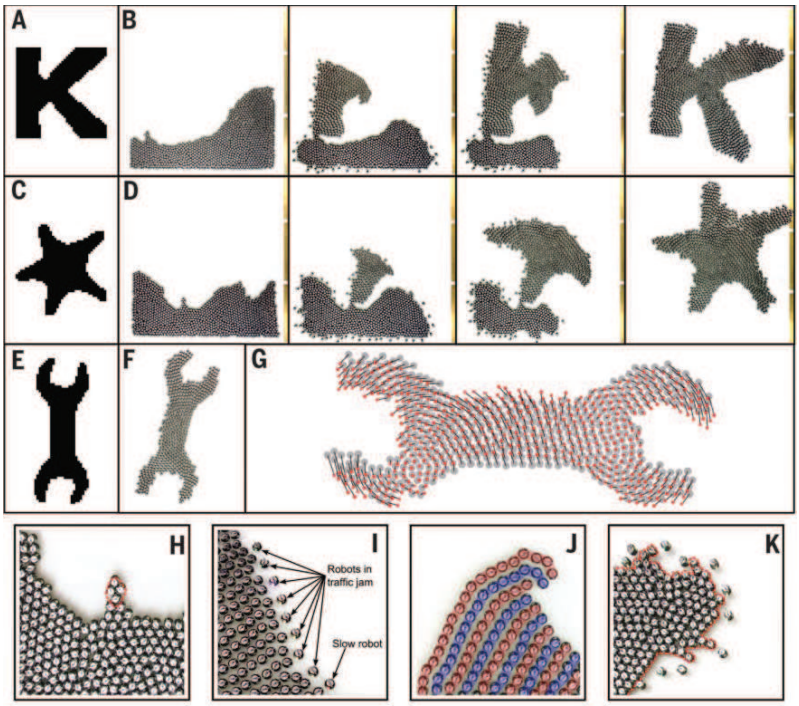


Fig. 2.9, Self-assembly experiments (Rubenstein, Cornejo, & Nagpal, 2014) using up to 1024 physical robots. (A, C, and E) Desired shape provided to robots as part of their program. (B and D) Self-assembly from initial starting positions of robots (left) to final self-assembled shape (right). Robots are capable of forming any simply connected shape, subject to a few constraints to allow edge-following (19). (F) Completed assembly showing global warping of the shape due to individual robot errors. (G) Accuracy of shape formation is measured by comparing the true positions of each robot (red) and each robot's internal localized position (gray). (H to K) Close-up images of starting seed robots (H), traffic backup due to a slowly moving robot (I), banded patterns of robots with equal gradient values after joining the shape (robots in each highlighted row have the same gradient value) (J), and a complex boundary formed in the initial group (dashed red line) due to erosion caused by imprecise edge-following (K).

2.3.2 Towards Real World Application

In sections 2.3 promising properties of swarm robotics have been enlightened. Nevertheless, currently, there exist no real commercial applications. The reasons for it are varied. Sahin and Winfield (Şahin E. & Winfield, 2008) enumerate three of them as follows.

1. *Algorithm Design*. Swarm robotics must design both the physical robots and the behaviors of the individual robots, so the global collective behavior emerges from their interactions. Right now, no general method exists to go from the individuals to the group behavior.
2. *Implementation and Test*. The use of many real robots needs of good laboratory infrastructure to be able to perform experiments.
3. *Analysis and Modeling*. Swarm-robotic systems are usually stochastic and nonlinear, so building mathematical models for validation and optimization is hard. These models might be necessary for creating safety real world applications.

Winfield et al. (Winfield, Harper, & Nembrini, 2005) discuss the concept of swarm engineering, studying the dependability of swarm-robotic systems through a case of study. Higgins et al. (Higgins, Tomlinson, & Martin, 2009.) address the main security challenges that swarm-robotic systems should face in a future. They state that due to the simplicity of swarm-robotic architectures they have to deal with the following problems:

1. Physical capture of the robots.
2. Identity and authentication, a robot must know if it is interacting with a robot from its swarm or from an intruder robot.
3. Communication attacks, communications can be intercepted or disturbed by an attacker.

The possible real applications of swarm robotics will take special importance when robots get to be mass produced and the costs of building swarms of robots decrease. This is the objective of I-swarm project (Seyfried, et al., 2005) which aimed at building a swarm of micro-robots. The development of technologies such as MEMS (Micro-Electro-Mechanical Systems) will allow creating small and cheap robots. Swarm robots can perform tasks in which the main goal is to cover a wide region. The robots can disperse and perform monitoring tasks, for example, in forests, lakes, and so forth. It can be really useful for detecting hazardous events, like a leakage of a chemical substance. The main advantage over a sensor network is that the swarm can move and focus on the problem and even act to prevent the consequences of that problem.

In this way, swarms of robots can be useful for dangerous tasks, for example for mining detection and cleaning. It can be more useful than a unique specialized robot, mainly because of the robustness of the swarm: if one robot fails and the mine explodes, the rest of the swarm continues working. In the case of a single robot, this is not possible. The number of possible applications is promising, but still the technology must firstly be developed both in the algorithmic and modeling part and in the miniaturization technologies.

3 Biologically Inspired Optimization Methods

"Everything you can imagine
nature has already created."
Albert Einstein

Optimization refers to the process of manipulating a computational structure or system to achieve some pre-specified goal. The systems under consideration can often be expressed in terms of mathematical function, and the goal is then to find the minimum or maximum of this function. Optimization plays a central role in science and engineering. There are so many examples of applications involving optimization that any list of such application is bound to be incomplete and biased. Nevertheless, some classes of problems where optimization is highly relevant include scheduling, decision making, microchip design, transportation-system design, various engineering problems and so forth.

For such a problem, all the tools of analysis, algebra and other sub-disciplines of mathematics can be summoned and, indeed, many optimization algorithms have been developed in mathematics. These algorithms define the field of **classical optimization**, and they are particularly well-developed for a class of optimization problems referred to as *convex optimization problems*. However, classical optimization algorithms also have some shortcomings and they are not suitable for all the optimization problems. Fortunately, there are plenty of alternative algorithms, most of which have been developed in the last few decades and are inspired by biological phenomena.

This chapter introduces several SI-based algorithms, highlighting their notable variants, their merits and demerits, and their applications. These algorithms include Genetic Algorithms (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Differential Evolution (DE), Artificial Bee Colony (ABC) and Glowworm Swarm Optimization (GSO).

All these algorithms are *stochastic* and *inspired by biological phenomena*. The stochasticity of the algorithms implies that different results may be obtained upon running such algorithms repeatedly. By contrast, the classical optimization algorithms are deterministic, so that, starting from a given initial condition, the results obtained are always

the same. To practitioners of classical optimization methods, the stochasticity of the results obtained may seem a clear disadvantage, but there are, in fact, advantages as well. An example is the ability of stochastic optimization algorithms to find several different, but equally, viable, solutions to problems, a property that is used in, for example, design optimization.

The second property, i.e. the inspiration taken from natural processes, may also require the reason why one should base an algorithm on biological phenomena. The answer is that Nature is all about ***adaptation***, which can be considered a form of optimization; the term adaptation refers to the gradual change in properties or behaviors to various in surrounding. Thus, for example, the growth of thick fur on artic animals is an adaptation to cold weather. Biological organisms display an amazing variety of adaptation. Another excellent example, relevant for engineering optimization, is shark skin. Sharks have been around for a long time; the ancestors of today's sharks appeared even before the dinosaurs. Thus, evolution, which forms the basis for the evolutionary algorithms, has had a very long time to work on the design of these remarkable animals. If one touches a shark, one can feel that its skin is not smooth, but covered with small ribs [Fig. 3.1] aligned with the direction of motion (Ball, 1999). These structures affect the interaction between the skin of the shark and the tiny fluid vortices that appear as a result of movement. The net effect is a reduction in friction which enables the shark to swim faster. Engineers are testing the effects of adding similar microstructures to the surfaces of ships and aircraft as well as the swimsuits of swimmers.

Obviously, sharks have not evolved their features in isolation: their prey has concurrently evolved various means of escaping, for example through coordinated swarm behavior. The simultaneous evolution of the properties of both predator and prey is an example of co-evolution, a phenomenon that has also been exploited about stochastic optimization (Hillis, 1990).

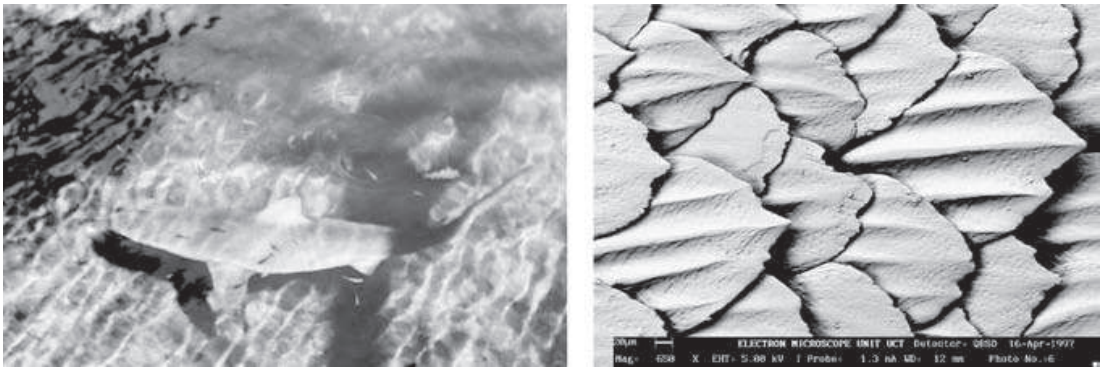


Fig. 3.1, The skin of a shark (right panel) contains small, riblike structures lying in the direction of motion that allow the animal to swim faster.

Another prime example is the evolution of the eye. As Dawkins (Dawkins, 1996) has noted, the eye has evolved in no less than 40 different ways, independent of each other. Two examples, shown in Fig. 3.2, are the compound eyes of insects and the lens eyes of mammals. Thus, evidently, there are many different solutions to the problem of generating a light-gathering device.

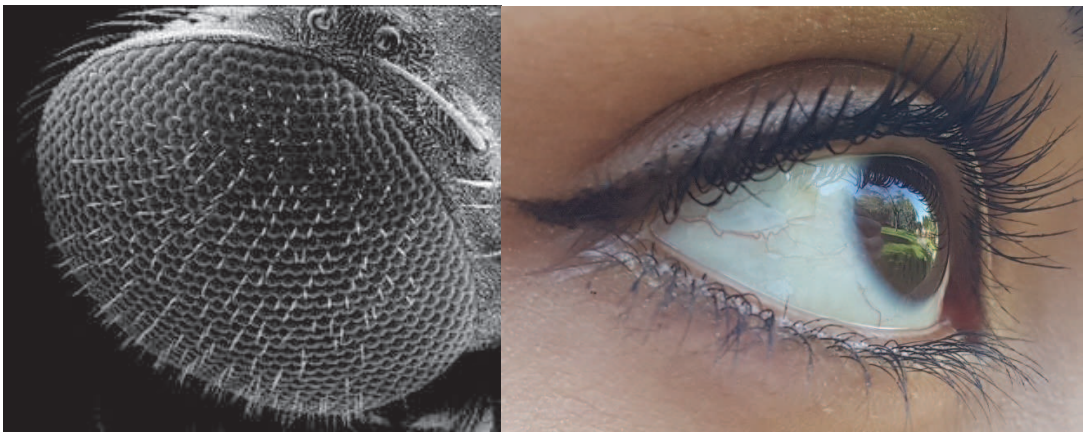


Fig. 3.2, Left panel: the compound eye of a fly. Right panel: a human eye. These are only two of the many light-gathering devices generated by biological evolution.

In addition to adaptation, many biological organisms also display ***cooperation***. Even though cooperation is omnipresent in nature, nowhere is it as prevalent as in certain species of insects, such as ants, bees, and termites. An example is the foraging behavior of ants. Using only local communication, ants are able to coordinate their food gathering with amazing efficiency, and their behavior forms the basis for ant colony optimization. Swarming behavior, as seen in flocks of birds or schools of fish, is another form of cooperation which serves several purposes, such as improving the efficiency of food gathering and protecting against predators. From the discussion above, it should be evident that there is ample motivation for basing optimization algorithms on biological phenomena.

However, before plunging into the field of optimization methods, it is mandatory to say a few words about the so-called **No-free-lunch** (NFL) **theorem** (Wolpert & Macready, 1997). The theorem concerns optimization algorithms based on search. In such algorithms, candidate solutions to the problem at hand are generated and evaluated one after another. Essentially, the implication of the NFL theorem is that, averaged over all possible problems, or objective functions, no search algorithm outperforms any other algorithm. This would imply, for example, that a completely random search would do just as well as any of the stochastic optimization algorithms considered in this chapter. Yet, while the NFL theorem is certainly valid, it is rarely of practical significance, as one never considers all possible

problems; in practice, it is found that, in a typical optimization problem, stochastic optimization algorithms such as genetic algorithms easily outperform random search. However, the theorem does imply that one should be careful before extrapolating the estimated performance of a stochastic optimization algorithm from one problem to another. Nevertheless, it is a fact that some algorithms do outperform others on specific classes of problems. For example, ant colony optimization, studied in Section 3.2, is particularly efficient when applied to problems involving path generation.

3.1 Genetic Algorithm

The Genetic Algorithm (GA) introduced by John Holland in 1975 (Goldberg, 1987), (Holland, 1992), is a search optimization algorithm based on the mechanics of the natural selection process. The basic concept of this algorithm is to mimic the concept of the *"survival of the fittest"*; it simulates the processes observed in a natural system where the strong tends to adapt and survive while the weak tends to perish.

GA is a population-based approach in which members of the population are ranked based on their solutions' fitness. A new population is formed using specific genetic operators such as *crossover*, *reproduction*, and *mutation*. The population can be represented in a set of strings (referred to as *chromosomes*). In each generation, a new chromosome (a member of the population) is created using information originated from the fittest chromosomes of the previous population. GA generates an initial population of feasible solutions and recombines them in a way to guide their search toward more promising areas of the search space.

Each of these feasible solutions is encoded as a chromosome, also referred to as *genotype*, and each of these chromosomes will get a measure of fitness through a fitness function (evaluation or objective function). The value of fitness function of a chromosome determines its capability to endure and produce offspring. The high fitness value indicates the better solution for maximization and the low fitness value shows the better solution for minimization problems. A basic GA has five main components: a random number generator, a fitness evaluation unit, a reproduction process, a crossover process, and a mutation operation. Reproduction selects the fittest candidates of the population, while crossover is the procedure of combining the fittest chromosomes and passing superior genes to the next generation, and mutation alters some of the genes in a chromosome.

Fig 3.3 shows the general flow chart of GA and the main components that contribute to the

overall algorithm. The operation of the GA starts with determining an initial population whether randomly or using some heuristics. The fitness function is used to evaluate the members of the population and then they are ranked based on the performances. Once all the members of the population have been evaluated, the lower rank chromosomes are omitted and the remaining populations are used for reproduction. This is one of the most common approaches used for GA. Another possible selection scheme is to use pseudo-random selection, allowing lower rank chromosomes to have a chance to be selected for reproduction. The *crossover* step randomly selects two members of the remaining population (the fittest chromosomes) and exchanges and mates them. The final step of GA is the *mutation*. In this step, the mutation operator randomly mutates on a gene of a chromosome. Mutation is a crucial step in GA since it ensures that every region of the problem space can be reached. *Elitism* is used to prevent the best solution of the population from being destroyed during crossover and mutation operation. Elitism guarantees the fitness of new generation will be at least as good as the current generation. The evaluation and generation of the new populations continue until the maximum number of generations is reached or the optimum solution is found. GA is advantageous in terms of requiring limited parameter settings and initializing itself from possible solutions rather than a single solution.

One of the main drawbacks of GA is the lack of fast convergence towards the optimal values since the crossover and mutation process are random. The applications of GA are wide ranging from scheduling, machine learning, robotics, signal processing, business, mathematics, manufacturing, routing, and much more.

Since the introduction of GA, many researchers have conducted studies to improve the performance of the GA. They have introduced several alternative approaches for crossover and mutation to enhance the quality of solutions. In crossover, instead of selecting one crossover point, De Jong et al. (1992) and Üçoluk (2002) have introduced *N-point crossover* and *segmented crossover* which selects several points for crossover (Üçoluk, 2002), (De Jong & Spears, 1992). The difference between them is *N*-point crossover is choosing several breaking points randomly, while in segmented crossover, only two breaking points are utilized. Mutation is one of the most important operators in GA to direct the chromosomes towards the better solution. Therefore, several studies have given different methods for mutation. By default, each gene in a chromosome is assigned with probability, p_m, and mutated depending on that probability. This mutation is known as uniform mutation. The other approaches for mutation are bitwise inversion where the whole gene in a chromosome

is mutated using a random mutation. Adaptive genetic algorithms have been introduced in order to allow the use of precise parameters in setting the population size, the crossing over probability, and the mutation probability. All of these parameters are dynamic and changing over the iterations. For instance, if the population is not improving, the mutation rate is increasing and whenever the population is improving, the mutation rate starts decreasing (Chiao Mei, Phon-Amnuaisuk, Alias, Leong, & Adaptive, 2008). Raja and Bhaskaran (Raja & Bhaskaran, 2013) have suggested a new approach of GA initialization that improve the overall performance of GA. In this approach, they utilized initialization twice where the first initialization is used to identify the promising area. After the first initialization, all chromosomes are ranked and the best chromosomes are selected. After that, GA has initialized again within the area where the best chromosomes have been identified.

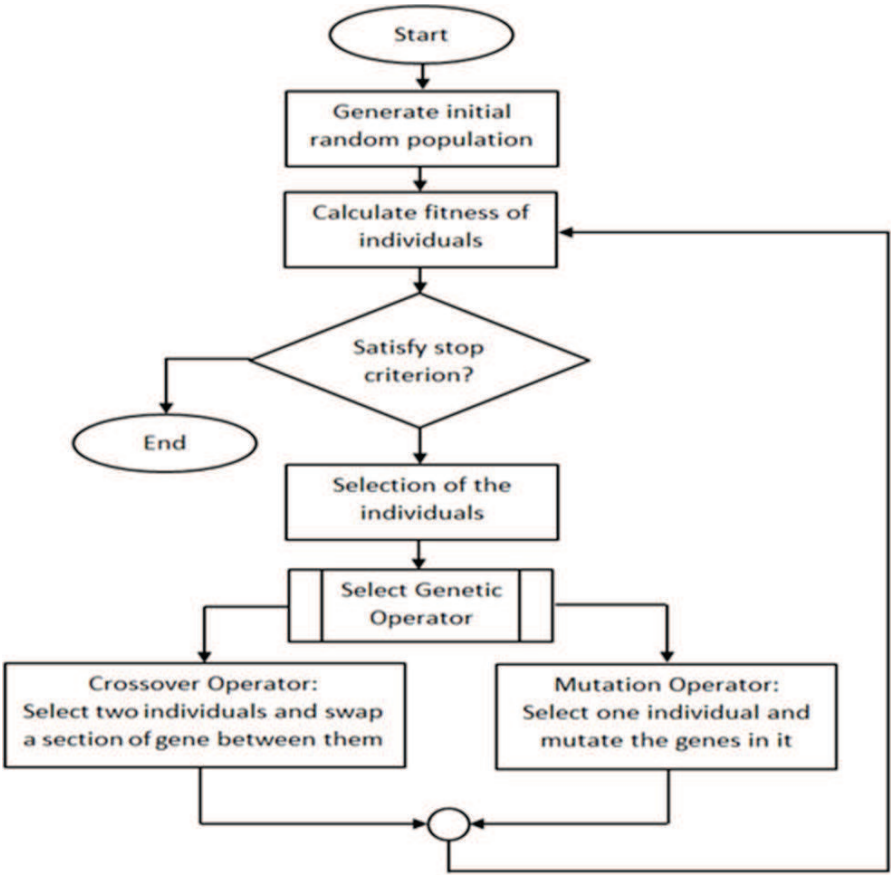


Fig. 3.3, Flow Chart of Genetic Algorithm with all steps involved from beginning until termination conditions met.

3.2 Ant Colony Optimization

Ant Colony Optimization (ACO) is a metaheuristic approach inspired by the Ant System (AS) proposed by Marco Dorigo in 1992 in his Ph.D. thesis (Dorigo M. , 1992), (Dorigo, Birattari, & Stutzle, 2006). It is inspired by the foraging behavior of real ants. This algorithm consists of four main components (ant, pheromone, daemon action, and decentralized control) that contribute to the overall system. Ants are imaginary agents that are used to mimic the exploration and exploitation of the search space. In real life, the pheromone is a chemical material spread by ants over the path they travel and its intensity changes over time due to evaporation. In ACO the ants drop pheromones when traveling in the search space and the quantities of these pheromones indicate the intensity of the trail. The ants choose the direction based on path marked by the high intensity of the trail. The intensity of the trail can be considered as a global memory of the system. Daemon actions are used to gather global information which cannot be done by a single ant and uses the information to determine whether it is necessary to add extra pheromone to help the convergence. The decentralized control is used to make the algorithm robust and flexible within a dynamic environment. The importance of having a decentralized system in ACO is due to resulting flexibility in the face of ant lost or ant failure offered by such a system. These basic components contribute to a cooperative interaction that leads to the emergence of shortest paths. Fig 3.4. depicts the initial phase, mid-range status of any system, and the final outcomes of the ACO algorithm respectively. The left figure illustrates the initial environment when the algorithm starts, where an ant starts moving randomly from the nest towards the source and returns. The middle figure illustrates several iterations of execution when ants discover multiple possible paths between nest and source. The shortest path is chosen, and ants use this path frequently which contributes to high intensity of pheromone trail as shown in the in Fig 3.4 (3).

N, *S*, *a*, and *b* represent nest, food source, on-going path, and returning path respectively. The steps involved to find the best solution starts with choosing the next node (from the current position in the search space) using following equation:

$$p_{(i,j)}^{k}(t) = \frac{\left\{\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}(t)\right]^{\beta}\right\}}{\sum_{k \in J_k}\left\{\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}(t)\right]^{\beta}\right\}} \tag{3.1}$$

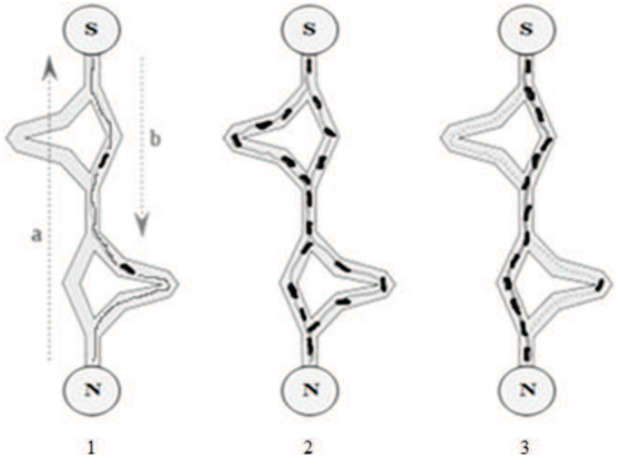


Fig. 3.4, Ant Colony Optimization Algorithm processes. *N* and *S* denote Nest and Source, while *a* and be respectively the ongoing and returning directions. Fig. 3.4 (1) shows early process where ants start to find a path between nest and source and lay pheromone. Fig. 3.4 (2) shows intermediate process where ants went through all possible paths. Fig. 3.4 (3) shows most of the ants choose path with the highest pheromone.

$p_{i,j}$ is the probability of going from node i to node j. J_k are the nodes that the ant is allowed to travel to from node i. η_{ij} contributes to the visibility between node i and node j. $\tau_{ij}(t)$ represents the amount of unevaporated pheromone between node i and node j at time t. α and β in Eq. 3.1 control the influence of τ_{ij} and η_{ij}, where if α is higher, the searching behavior of ant is more depending on pheromone and if β is higher, the searching behavior of ant is depending on its visibility or knowledge. Each ant also has a taboo list which is used to prevent any ants from visiting the same node twice.

Pheromones, as stated before, are one of the crucial components in ACO as they leave trails which increase the probability of the next ant choosing the same path. In order to deposit a pheromone, the following equation is used:

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L_k}(t) \\ 0 \end{cases} \tag{3.2}$$

Q is a constant, L is the cost of the ant's tour, (i.e., the length of the generated path), t is the iteration number and k represents a specific ant. The value represents the pheromone rate between node i and node j that the ant visited in iteration t. The pheromone deposition value for a path that is not selected is zero. Another important component is the pheromone evaporation rate. This component determines the exploration and exploitation behavior of the ant. High and low evaporation rates result in exploration and exploitation behaviors respectively. Too high exploration rates result in ants getting lost, while too low values result in an inability to acquire the optimal path (Dorigo M. , 1992). The pheromone decay factor is utilized using following equation:

$$\tau_{(i,j)}(t+1) = (1-p) \cdot \tau_{(i,j)}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \quad (3.3)$$

m is the number of ants in the system and p is the pheromone evaporation rate or decay factor. ACO has several advantages over other evolutionary approaches including offering positive feedback resulting in rapid solution finding and having distributed computation which avoids premature convergence. These are in addition to taking advantage of the existing collective interaction of a population of agents. However, ACO has drawbacks such as slower convergence compared with other heuristic-based methods and lack a centralized processor to guide it towards good solutions. Although the convergence is guaranteed, the time for convergence is uncertain. Another important demerit of ACO is its poor performance within problems with large search spaces. ACO has been applied in various optimization problems such as traveling salesman problem (TSP) (Valdez & Chaparro, 2013), quadratic assignment problem (Tosuna, Dokeroglua, & Cosara, 2013), vehicle routing (Yagmahan & Yenisey, 2010) and so on. Several ACO variants have been created with the aim to improve overall performance.

Two years after the introduction of ACO, Dorigo and Gambardella made modifications by improving three major aspects (pheromone, state transition rule and local search procedures) which produce the variant of ACO called Ant Colony System (ACS) (Dorigo & Gambardella, 1997). ACS uses centralize (global) update approach for pheromone update and only concentrate the search within a neighborhood of the best solution found so far in order to increase efficiency for convergence time. The state transition rule is different from ACO where ACS has a stated probability (q_0) to decide which behavior is used by the ant. q_0 is usually set to 0.9 and compare to a value of q (which $0 \leq q \leq 1$). If the value of q is less than that, then exploitation behavior is used and vice versa. For local search procedures, a local optimization heuristic based on an edge exchange strategy such as 2-opt, 3-opt or Lin-Kernighan is applied to each solution generated by an ant to get its local minima. This combination of new pheromone management, new state transition, and local search procedures has produced a variant of ACO for TSP problems. Max-Min Ant System (MMAS) is considered as another notable variant of ACO. The approach was introduced by Stutzle and Hoos in 2000 and it limits the pheromone trail values within the interval of $[\tau_{min}, \tau_{max}]$ (Stützle & Hoos, 2000). MMAS also modified three aspects of ACO. First, at the beginning, the pheromone trails are set to the maximum value which escalates the exploration behavior of the ants. Second, the authors introduce an interval of $[\tau_{min}, \tau_{max}]$ which limits the pheromone trails in order to avoid stagnation. Third, only one ant is allowed

to add pheromone which helps to exploit the best solutions found during the execution of the algorithm. The pheromone may be added by using either an iteration-best approach or a global-best approach. In the iteration-best approach, only the ant with the best solution adds the pheromone for each iteration while in the global-best approach, the ant with the best solution can add the pheromone without considering other ants in the same iteration.

3.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an optimization technique introduced by Kennedy and Eberhart in 1995 (Kennedy & Eberhart, 1995). It uses a simple mechanism that mimics swarm behavior in birds flocking and fish schooling to guide the particles to search for global optimal solutions. Del Valle and his co-authors (DelValle, Venayagamoorthy, Mohagheghi, Hernandez, & Harley, 2008) described PSO with three simple behaviors of *separation*, *alignment*, and *cohesion* as shown in Fig 3.5 respectively. Separation is the behavior of avoiding the crowded local flockmates while alignment is the behavior of moving towards the average direction of local flockmates. Cohesion is the behavior of moving towards the average position of local flockmates. The formulas of PSO algorithm are as follows:

$$v_{id}^{t+1} = v_{id}^{t} + c_1 \cdot rand(0,1) \cdot (p_{id}^{t} - x_{id}^{t}) + c_2 \cdot rand(0,1) \cdot \left(p_{gd}^{t} - x_{id}^{t}\right) \tag{3.4}$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \tag{3.5}$$

where v_{id}^{t} and x_{id}^{t} are particle velocity and particle position respectively. d is the dimension in the search space, i is the particle index, and t is the iteration number. c_1 and c_2 represent the speed, regulating the length when flying towards the most optimal particles of the whole swarm and the most optimal individual particle. p_i is the best position achieved so far by particle i and p_g is the best position found by the neighbours of particle i. $rand(0,1)$ is the random values between 0 and 1. The *exploration* happens if either or both of the differences between the particle's best (p_{id}^{t}) and previous particle's position (x_{id}^{t}), and between population's all-time best (p_{gd}^{t}) and previous particle's position (x_{id}^{t}) are large, and *exploitation* occurs when these values are both small. PSO proved to be an efficient optimization algorithm by searching an entire high-dimensional problem space. It is a robust stochastic optimization technique based on the movement and intelligence of swarms. It applies the concept of social interaction to problem-solving and does not use the gradient of the problem being optimized, so it does not require the optimization problem to be

differential, as is required by classic optimization methods (Yan, Wu, Liu, & Huang, 2013). The optimization of irregular problems that are noisy and change over time can be determined using PSO (Arumugam, Rao, & Tan, 2009:). The parameters of PSO consist of a number of particles, position of agent in the solution space, velocity and neighborhood of agents (communication of topology).

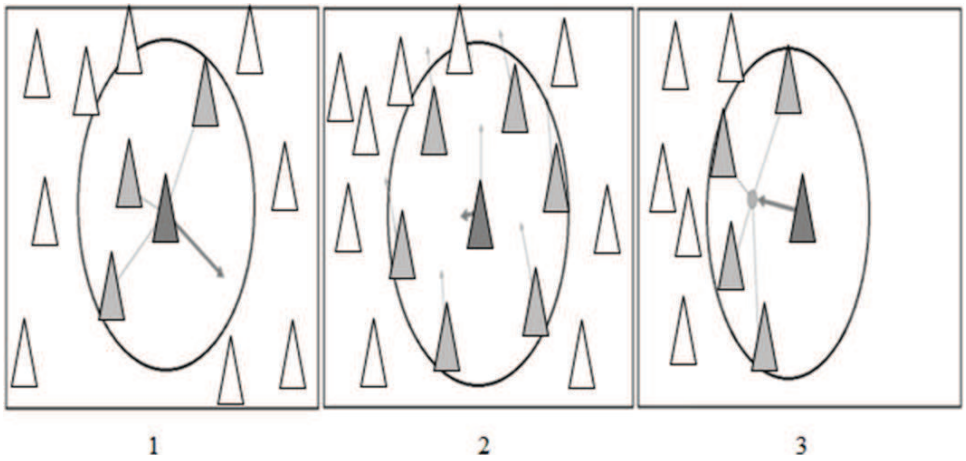


Fig 3.5, PSO Basic Behaviors. Figure 3.5 (1) shows separation behavior where particle avoiding other particles. Figure 3.5 (2) shows alignment behavior where particle moving towards head of local flockmates and maintain the speed between them. Figure 3.5 (3) shows cohesion behavior where particle moving towards the average position of local flockmates (Yagmahan & Yenisey, 2010)**.**

The PSO algorithm begins by initializing the population first. The second step is calculating the fitness values of each particle, followed by updating individual and global bests, and later, the velocity and the position of the particles get updated. The second to fourth steps get repeated until the termination condition is satisfied (DelValle, Venayagamoorthy, Mohagheghi, Hernandez, & Harley, 2008). Fig 3.5 illustrates the PSO algorithm output over iterations. In the first iteration, all particles spread out in order to find the best solution (exploration). Each particle is evaluated. The best solutions are found with respect to neighborhood topology and the personal and global best particles for each member of the swarm are updated. The convergence would be achieved through attracting all particles towards the particle with the best solution.

The PSO algorithm has many merits. It is simple to implement, has only a few parameters to be set, it is effective in global search, it is insensitive to scaling of design variables, and it is easily parallelized for concurrent processing (Poli, Kennedy, & Blackwell, 2007). PSO has tendency to result in a fast and premature convergence in mid optimum points, in addition to having slow convergence in a refined search area (having weak local search ability) (Bai, 2010). PSO is used in networking, power systems, signal processing, control system, machine learning, image processing, and much more.

There are several approaches that can be used to improve PSO in general. The size of the population is one of the important factors. Higher population size can increase the chance of faster and precise convergence. A second approach is to achieve a balance between exploration and exploitation. In the beginning of iteration, high exploration would give a high chance to find a solution which is close to global optima. Meanwhile, towards the end of iteration, high exploitation would give a chance for particle to find the most accurate solution within the promising area. A sub-swarm approach is another way that can be used to increase the basic PSO performance which is quite commonly used nowadays. Allocating different tasks or objectives to each sub-swarm can also increase the efficiency of PSO in the multi-objective problems. Another approach to improve the PSO performance is to set the contributing components of the velocity equation (dynamic velocity adjustment). Such an approach can direct particles in different directions resulting in faster convergence towards a global optimum (Atyabi & Powers, 2013).

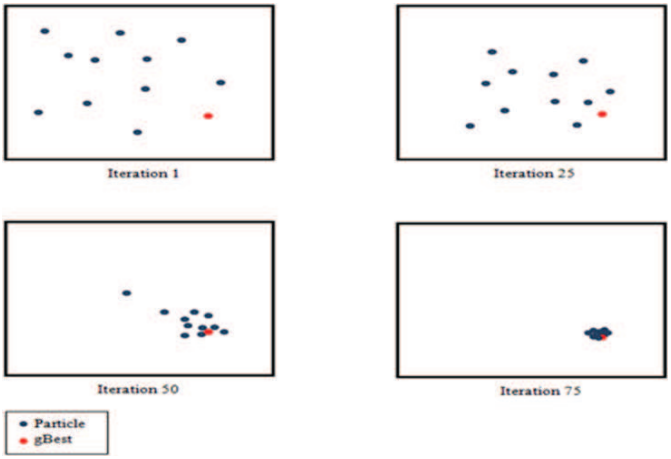


Fig 3.6, Particle Swarm Optimization movement towards global optima over iteration numbers.

The two most notable variants in PSO are the introduction of inertia weight and constriction factors. Inertia weight (w) is introduced by Shi and Eberhart three years after PSO was first introduced to regulate the influence of the previous velocity which also controls the exploration and the exploitation behaviors of particle (Shi & Eberhart, 1998). If the w value is high then the step size is big, resulting in the occurrence of exploration behavior. If the w value is low then the step size is small and the exploitation behavior occurs. This element has been accepted as new standard form of velocity equation for basic PSO as illustrated in Eq (3.6):

$$v_{id}^{t+1} = w \cdot v_{id}^{t} + c_1 \cdot rand(0,1) \cdot (p_{id}^{t} - x_{id}^{t}) + c_2 \cdot rand(0,1) \cdot \left(p_{gd}^{t} - x_{id}^{t}\right) \qquad (3.6)$$

The introduction of inertia weight has improved overall performance of PSO in terms of the speed of convergence and the quality of solutions. From there, much research has been done to find the best configuration for inertia weight in order to optimize the convergence speed and the solutions' quality. Bratton and Kennedy suggested to use an inertia weight value higher than 1.0 and decreasing eventually to a value lower than 1.0 with the aim of encouraging exploration at an early stage and exploitation of the best area found towards the end (Bratton & Kennedy, 2007). Clerc and Kennedy later introduced the constriction factor named as K in order to increase the chance of convergence and avoid particles from leaving the search space (Clerc & Kennedy, 2002).

$$v_{id}^{t+1} = \mathrm{K}\left[v_{id}^{t} + c_1 \cdot rand(0,1) \cdot (p_{id}^{t} - x_{id}^{t}) + c_2 \cdot rand(0,1) \cdot \left(p_{gd}^{t} - x_{id}^{t}\right)\right] \qquad (3.7)$$

Both variants have improved the overall performance of the PSO algorithm. Eberhart and Shi have compared these two variants and come to the conclusion that the constricted PSO perform better than the improved basic PSO (Eberhart & Shi, 2000).

3.4 Differential Evolution

The Differential Evolution (DE) algorithm is a population-based algorithm that can be considered to be similar to GA since it employs similar operators: *crossover*, *mutation*, and *selection*. The main difference between DE and GA is in constructing better solutions, where DE relies on mutation operation while GA relies on crossover operation. This algorithm was introduced by Storn and Price in 1997 (Storn & Price, 1997). Since this algorithm relies on mutation operation, it utilizes the mutation as a search mechanism and takes advantage of the selection operation in order to direct the search towards the potential regions in the search space.

Target Vector, *Mutant Vector*, and *Trail Vector* are three properties that DE utilizes for generating a new population iteratively. The target vector is the vector that contains the solution for the search space, the mutant vector is the mutation of the target vector, and the trail vector is the resultant vector after the crossover operation between target vector and mutant vector. The basic steps of the DE algorithm as stated before, are similar to GA with only slight differences (Price, Storn, & Lampinen, 2005). DE starts with steps such as population initialization followed by evaluation to determine the fittest members of the population. Later, new parameter vectors get generated by adding the weighted difference of the two population vectors with the third vector. This step is referred to as mutation.

Within the crossover, the vector is mixed and the algorithm takes a final step of selection. In order to see the differences between DE and GA, a more detailed discussion on the three main operators in DE is required.

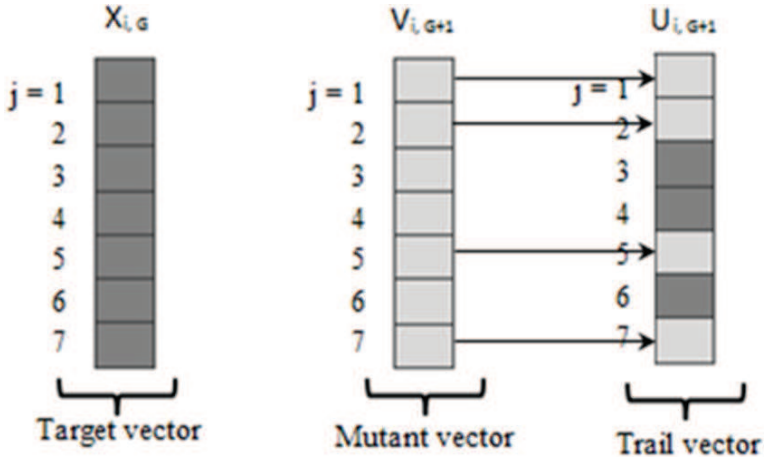


Fig. 3.7, Illustration of Crossover Process of DE with vector dimension (j) of 7. Target vector is current solution with mutant vector is another possible solution. Trail vector is new solution after crossover process between target vector and mutant vector.

In the mutation step, each of N parameter vectors goes through mutation. Mutation is the operation of expanding the search space and a mutant vector is generated by:

$$v_{i,G+1} = x_{r1,G} + \mathrm{F}\left(x_{r2,G} - x_{r3,G}\right) \tag{3.8}$$

where F is the scaling factor with a value in the range of $[0,1]$ with solution vectors x_{r1}, x_{r2}, and x_{r3} being chosen randomly and satisfying following criteria:

$$x_{r1}, x_{r2}, x_{r3} \,|_{r_1 \neq r_2 \neq r_3 \neq i} \tag{3.9}$$

where i is the index of the current solution. Fig 3.7 illustrates a two-dimensional vector which plays a part in generating the mutant vector. Crossover operation is introduced to increase the diversity of the disconcerted parameter vectors. The parent vector is mixed with a mutated vector and a trial vector is produced by:

$$u_{i,G+1} = \begin{cases} v_{i,G+1} \; if \; R_j \leq CR \\ x_{i,G} \; if \; R_j > CR \end{cases} \tag{3.10}$$

where CR is a crossover constant and R_j is a random real number between $[0,1]$ with j denoting the j^{th} component of the resultant array.

In DE, all solutions in the population have the same probability of being selected as parents without considering their fitness value. This is the main difference in the operations of DE and GA. Simply put, the child (trail vector) produced is only evaluated after mutation and crossover operations. After that, the performance of this child vector is compared to its parent and the better vector is retained in the population. The exploitation behavior occurs when

the difference between two solution vectors in Eq. 3.8 are small, while the exploration behavior occurs when the difference between those two are large.

DE is advantageous in terms of enhancing the capacity of local search and keeping the multiplicity of the population while it suffers from slow convergence and being unstable (Wu, Lee, & Chien, 2011). DE is employed in various applications such as electrical engineering, image processing, machine learning, and economy.

3.5 Artificial Bee Colony

Artificial Bee Colony (ABC) is one of the most recent swarm intelligence algorithms. It was proposed by Dervis Karaboga in 2005 (Karaboga, 2005); in 2007, the performance of ABC was analyzed (Karaboga & Basturk, 2007) and it was concluded that ABC performs quite well compared with several other approaches.

This algorithm is inspired by the intelligent behavior of real honey bees in finding food sources, known as nectar, and the sharing of information about that food source among other bees in the nest. This algorithm is claimed to be as simple and easy to implement as PSO and DE (Karaboga D., 2010). In this approach, the artificial agents are defined and categorized into three types, the *employed bee*, the *onlooker bee*, and the *scout bee*. Each of these bees has different tasks assigned to them in order to complete the algorithm's process. The employed bees focus on a food source and retain the locality of that food source in their memories. The number of employed bees is equal to the number of food sources since each employed bee is associated with one and only one food source. The onlooker bee receives the information of the food source from the employed bee in the hive. After that, one of the food sources is selected to gather the nectar. The scout bee is in charge of finding new food sources and the new nectar. The general process of ABC method and the details of each step are as follows (Karaboga, 2005):

Step 1. Initialization Phase: All the vectors of the population of food source, $\vec{x}_i$, are initialized ($i = 1 \ldots SN$, where SN is population size) by scout bees and control parameters being set. Each $\vec{x}_i$ vector holds n variables, which is optimized, to minimize the objective function. The following equation is used for initialization phase:

$$x_i = l_i + rand(0,1) * (u_i - l_i) \tag{3.11}$$

where l_i and u_i respectively are the lower and upper bound parameters of x_i.

Step 2. Employed Bees Phase: In this phase, the search for a new food source, $\vec{v}_l$, increases in order to have more nectar around the neighborhood of the food source, $\vec{x}_l$. Once a neighboring food source is found, its profitability or fitness is evaluated. The new neighboring food source is defined by using following formula:

$$v_i = x_i + \Phi_i(x_i - x_j) \tag{3.12}$$

where x_j is a randomly selected food source and Φ_i is a random number of $[-a, a]$. Once the new food source, v_i, is produced its profitability is measured and a greedy selection is applied between $\vec{x}_i$ and $\vec{v}_i$. The exploration happens if the difference between $x_i - x_j$ is large and the exploitation behavior is when the difference is small. The fitness value of the solution, $fit_i(\vec{x}_i)$, is determined using following equation:

$$fit_i(\vec{x}_i) = \begin{cases} \dfrac{1}{1 + f_i(\vec{x}_i)} & if\ f_i(\vec{x}_i) \geq 0 \\ 1 + abs\left(f_i(\vec{x}_i)\right) & if\ f_i(\vec{x}_i) < 0 \end{cases} \tag{3.13}$$

where $fit_i(\vec{x}_i)$ is the objective function value of solution $\vec{x}_i$.

Step 3. Onlooker Bees Phase: Onlooker bees that are waiting in the hive choose their food sources depending on probability values measured using the fitness value and the information shared by employed bees. The probability value, p_i, is measured using the following equation:

$$p_i = \frac{fit_i(\vec{x}_i)}{\sum_{i=1}^{SN} fit_i(\vec{x}_i)} \tag{3.14}$$

Step 4. Scout Bees Phase: The scout bees are those unemployed bees that choose their food sources randomly. Employed bees whose fitness values cannot be improved through predetermined number of iterations, called as limit or abandonment criteria become the scout bees and all their food sources get abandoned.

Step 5. The best fitness value and the position associated with that value are memorized.

Step 6. Termination Checking Phase: If the termination condition is met, the program terminates, otherwise the program returns to Step 2 and repeats until the termination condition is met.

Advantages of ABC include being easy to implement, robust, and highly flexible. It is considered as highly flexible since only requires two control parameters of maximum cycle number and colony size. Therefore, adding and removing bee can be done without need to reinitialize the algorithm. It can be used in many optimization problems without any modification, and it requires fewer control parameters compared with other search

techniques. The disadvantages of ABC include the requirement of new fitness tests for the new parameters to improve performance, being quite slow when used in serial processing and the need for a high amount of objective function evaluations.

ABC has been implemented in various fields including engineering design problems, networking, business, electronics, scheduling and image processing.

Although ABC algorithm was only being introduced less than ten years ago there are already quite a number of variants of ABC available. One of the important ABC variants is Interactive ABC (IABC) designed to solve numerical optimization problems (Bolaji, Khader, Al-Betar, & Awadallah, 2013). Bao and Zeng have introduced three selection strategies of food source by onlooker bees for ABC which form three variants called Rank Selection Strategies ABC (RABC), Tournament Selection ABC (TABC) and Disruptive Selection ABC (DABC) (Bao & Zeng, 2009). The main aim for all these variants is to upsurge the population diversity and avoid premature convergence. Bao and Zeng have tested these modified ABCs with the standard ABC and the results showed that these three selection strategies perform better search compared with the standard ABC.

3.1 Glowworm Swarm Optimization

Glow worm Swarm Optimization (GSO) is a new SI-based technique aimed to optimize multimodal functions, proposed by Krishnanad and Ghose in 2005 (Krihnanand & Ghose, 2009), (Krihnanand & Ghose, 2009). GSO employs physical entities (agents) called glowworms. A condition of glowworm m, at time t has three main parameters of a position in the search space $\left(x_m(t)\right)$, a luciferin level $\left(l_m(t)\right)$ and a neighborhood range $\left(r_m(t)\right)$. These three parameters change over time. Initially, the glowworms are distributed randomly in the workspace, instead of finite regions being randomly placed in the search area as demonstrated in ACO. Later, other parameters are initialized using predefined constants. Yet, similar to other methods, three phases are repeated until the termination condition is satisfied. These phases are luciferin level update, glowworm movement, and neighborhood range update (Krihnanand & Ghose, 2009). In order to update the luciferin level, the fitness of current position of a glowworm m is determined using following equation:

$$l_m(t) = (1 - p) \cdot l_m(t - 1) + \gamma J\left(x_m(t)\right) \tag{3.15}$$

where p is the luciferin evaporation factor, γ is the luciferin constant and J is an objective function. The position in the search space is updated using following equation:

$$x_m(t) = x_m(t-1) + s\left(\frac{x_n(t-1) - x_m(t-1)}{\|x_n(t-1) - x_m(t-1)\|}\right) \tag{3.16}$$

where s is the step size, and ||.|| is Euclidean norm operator. If the difference between x_n and x_m is large then exploration behavior takes place and if this difference is small then exploitation behavior occurs. Later, each glowworm tries to find its neighbors. In GSO, a glowworm m is the neighbor of glowworm n only if the distance between them is shorter than the neighborhood range $r_m(t)$, and on condition where glowworm n is brighter than glowworm m. However, if a glowworm has multiple choices of neighbors, one neighbor is selected using the following probability equation.

$$p_m(t) = \frac{l_m(t) - l_n(t)}{\sum_{k \in Ni(t)} l_m(t) - l_n(t)} \tag{3.17}$$

where the probability of glowworm at m moving towards glowworm at n is the difference of luciferin level between them over difference of luciferin level between all glowworms within the range of glowworm m. The solution with the highest probability is selected and then the glowworm moves one step closer in direction of the chosen neighbor with a constant step size s. In the final phase, the neighborhood range $\left(r_m(t)\right)$ is updated to limit the range of communication in a group of glowworms. The neighborhood range is calculated using following equation:

$$r_m(t+1) = \min\{r_s, \max[0, r_m(t) + \beta(n_d - |n_m(t)|)]\} \tag{3.18}$$

where r_s is a sensor range (a constant that limits the size of the neighborhood range), $n_d$is the desired number of neighbors, $|n_m(t)|$ is a number of neighbors of the glowworm m at time t and β is a model constant. Fig. 3.8 illustrates two possible circumstances in GSO's agents' evolving procedures in which with respect to agents' positions in the search space and the available neighboring agent's different behaviors occurs. In (a), i, j and k represent the agents of glowworm. r_s^j denotes the sensor range of agent j and r_d^j denotes the local-decision range for agent j. The same applies with i and k where sensor range and local-decision range are represented by r_s^i and r_d^i and $r_s^k$and r_d^k respectively. It is applied in the circumstances where agent i is in the sensor range of agent j and k. Since the agents have different local-decision domains only agent j uses the information from agent i. In (b), a, b, c, d, e, and f are glowworm agents. 1, 2, 3, 4, 5, and 6 represent the ranking of the glowworm agents based on their luciferin values. Agents are ranked based on their luciferin values resulting in agent a being ranked 1 since it has the highest luciferin value.

GSO is effective within applications with limited sensor range and can detect multiple sources and is applicable to numerical optimization tasks. However, it also has low accuracy and slow convergence rate (Zainal, Zain, Radzi, & Udin, 2013). GSO has been applied to routing, swarm robotics, image processing, and localization problems.

GSO can be improved in general by considering the following modifications.

1) Expanding the neighborhood range to include all agents. Once the best solution has been determined, all agents can move towards the agent with the best solution. This step can increase the efficiency in exploitation since higher number of agents to be within the best solution range.
2) In order to increase GSO's convergence rate, the number of neighbors considered within the neighborhood range need to be as small as possible. This step might reduce the time taken for GSO since less calculation required to determine the probability and direction of its movement.

GSO has several variants that improve the overall performance of GSO. For example, He et al. (He, Tong, & Huang, 2012) introduced Improved GSO (IGSO) to take advantage of integrating chaos behavior in order to avoid local optima and increasing the speed and accuracy of convergence. Zhang et al. (Zhang, Ma, Gu, & Miao, 2011) have proposed two ideas to improve the performance of GSO. First, they proposed several approaches to alter the step-size of the glowworm such as fixed step, dynamic linear decreasing, and dynamic non-linear decreasing. Secondly, they proposed self-exploration behavior for GSO.

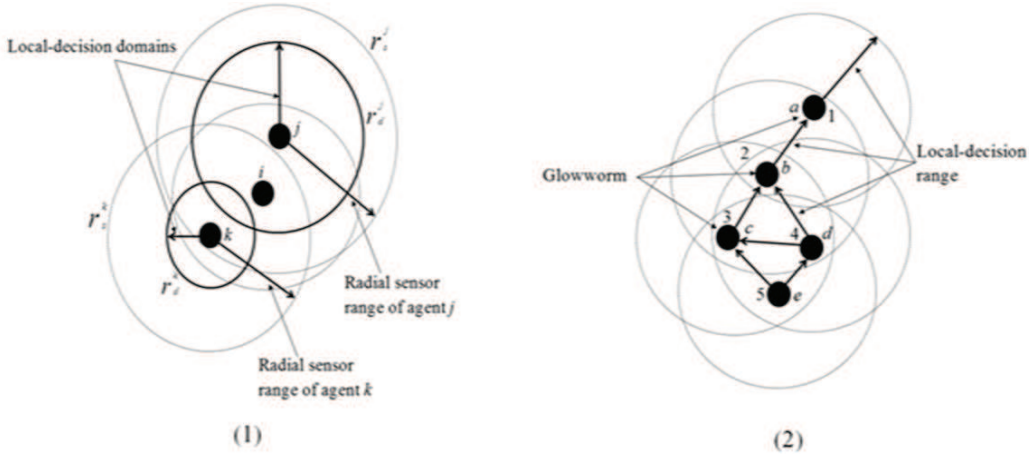


Fig. 3.8, Glowworm Search Optimization (GSO) in two possible conditions. a, b, c, d, e, f, i, j, and k are the glowworm agents. In subfigure (1), figure illustrates three glowworm agents with different sensor range and local-decision range. It shows if agent within local-decision of another agent, the agent with lower luciferin values move towards agent with higher luciferin values. In subfigure (2), glowworm agents are ranked based on their luciferin values with lower number represent higher luciferin values and higher number represent lower luciferin values.

4 Decision Making Model of Human Groups

"It is possible to make things of great complexity
out of things that are very simple.
There is not conservation of simplicity."
Stephen Wolfram

The second chapter pointed out on the ability of natural (Conradt 2012; Couzin 2009), (Conradt, 2003), (Couzin, 2005), and artificial, (Krieger M. J., 2000), (Rubenstein M. C., 2014), (Werfel, 2014), (Brambilla, 2013), groups in solving complex problems exceeding individual skills, focusing on the mechanisms behind the collective behaviors and introducing the concept of swarm intelligence (Vanni, Luković, & Grigolini, 2011).

The superior ability of groups in solving tasks originates from *collective decision making*: agents (animals, robots, humans) make choices, pursuing their individual goals (forage, survive, etc.) based on their own knowledge and amount of information (position, sight, etc.), and adapting their behavior to the actions of the other agents. The group-living enables social interactions to take place as a mechanism for knowledge and information sharing, (Clément, 2013), (Couzin, 2005), (Sumpter et al., 2009), (Ward et al., 2008), (Arganda et al., 2012), (Ward et al., 2011), (Perez-Escudero, 2011), (Watts, 2002), (Turalaska et al., 2009), (Wang & Szolnoki, 2013). Even though the single agents possess a limited knowledge, and the actions they perform usually are very simple, the collective behavior, enabled by the social interactions, leads to the emergence of a superior intelligence of the group.

In this chapter, the author introduces a new decision making model (DMM), firstly proposed by Carbone and Giannoccaro (Carbone & Giannoccaro, 2015) for solving complex combinatorial problems, showing a detailed analysis and understanding of its features and potentialities.

The DMM attempts to capture the main drivers of the individual behaviors in groups, i.e., *self-interest* and *consensus seeking*: individuals make choices based on rational calculation and self-interested motivations. Agent's choices are made by optimizing the perceived fitness value, which is an estimation of the real fitness value based on the level of agent's knowledge (Conradt, 2003), (Turalska & West, 2014), (Conradt, 2012). However, any

decision made by an individual is influenced by the relationships he/she has with the other group members. This social influence pushes the individual to modify the choice he/she makes, for the natural tendency of humans to seek consensus and avoid conflict with people they interact with (Di Maggio & Powell, 1983).

The Ising-Glauber dynamics (Castellano et al., 2009), (Glauber, 1963) is used to model the social interactions among group members. The *NK* model (Kauffman & Levin, 1987) (Kauffman & Weinberger, 1989) is employed to build the fitness landscape associated with the problem to solve. A continuous-time Markov chain governs the decision-making process, whose complexity is controlled by the parameter *K*.

The transition rate of individual's opinion change is defined as the product of the Ising-Glauber rate (Glauber, 1963), which implements the consensus seeking, (Sornette, 2014), (Weidlich, 1971), (Ising, 1925), (Brush, 1967), and an exponential rate, (Weidlich, 1991), (Schweitzer, 2007), which speeds up or slows down the change of opinion, to model the rational behavior of the individual.

We explore how both the strength of social interactions and the members' knowledge level influence the group performance. We identify in which circumstances human groups are particularly effective in solving complex problems. We extend previous studies highlighting the efficacy of collecting decision-making in presence of a noisy environment, (Gruènbaum, 1998), and in conditions of cognitive limitations, (Couzin, 2005), (Laughlin P. R., 2006), (Laughlin P. e., 2003), (Faria, 2009).

4.1 The model

Here the author presents the decision-making model (DMM) (Carbone & Giannoccaro, 2015). We consider a set of M interacting agents, which is assigned to carry out a complex task. The task consists in solving a combinatorial problem by identifying the set of choices, i.e. the set of N binary decisions, with the highest fitness, out of 2^N configurations. More precisely, each decision d_i of the bitstring $\boldsymbol{d} = (d_1, d_2, \dots, d_N)$ is a binary variable $d_i = \pm 1$, $i = 1, 2, \dots, N$. Each bitstring $\boldsymbol{d}$ is associated with a certain fitness value $V(\boldsymbol{d})$. The discrete landscape $V(\boldsymbol{d})$ must be properly chosen to make the optimization combinatorial problem belong to the class of the *NP*-complete problems. Different choices are possible, e.g. the fitness landscape can be represented by the length of the Hamiltonian cycle in the travelling

salesman problem (TSP), Knapsack problem, the Kauffman *NK* landscape, or any other *NP*-complex landscape (Garey & Johnson, 1979).

In this study we will exploit the complex landscape defined within the framework of the *NK* Kaufmann's model of combinatorial complexity (Kauffman & Levin, 1987), (Kauffman & Weinberger, 1989), (Weinberger & others, 1996), as this type of landscape makes it easy to model the level of knowledge of each agent in the groups, i.e. to take into account each agent in the group has a personal understanding of the problem.

Within the *NK* approach the discrete fitness function $V(\boldsymbol{d})$ is computed as the weighted sum of N independent stochastic contributions $W_j(\boldsymbol{d}_j^K)$, $j = 1, 2, \dots, N$ which only depend on the corresponding sub-bitstring $\boldsymbol{d}_j^K = \left(d_j, d_{j1}, \dots, d_{jK}\right)$ of length $K + 1$, where the K may take values $K = 0, 1, \dots, N - 1$ (Kauffman & Levin, 1987), (Kauffman & Weinberger, 1989), (Weinberger & others, 1996). The number of different values that each contribution $W_j(\boldsymbol{d}_j^K)$, may take is 2^{K+1}, i.e. it is equal to the number of different states that can be enumerated with bitstring $\boldsymbol{d}_j^K$. The fitness landscape $V(\boldsymbol{d})$ is then defined as

$$V(\boldsymbol{d}) = \frac{1}{N}\sum_{j=1}^{N} W_j(\boldsymbol{d}_j^K) \tag{4.1}$$

The integer index $K = 0, 1, 2, \dots, N - 1$, which provides the number of the interacting decision, tunes the complexity of the problem: increasing K increases the complexity of the problem. For $K \geq 2$ finding the optimum on an *NK* Kaufmann landscape is classified as an *NP*-complete problem. For more details on the Kaufmann *NK* model, the reader is referred to Appendix 8.1.

Each member of the group makes his/her choices driven by the rational behavior, which pushes him/her to increase the self-interest, and by social interactions, which push the member to seek consensus within the group.

To model the level of knowledge of the agents, we identify a parameter $p \in [0, 1]$, i.e. the probability that each single agent knows the contribution $W_j(d_j)$ to the total fitness. Based on the level of knowledge, each agent k computes his/her own perceived fitness (self-interest) as follows:

$$V_k(\boldsymbol{d}) = \frac{\sum_{j=1}^{N} D_{kj} W_j(\boldsymbol{d}_j^K)}{\sum_{j=1}^{N} D_{kj}} \tag{4.2}$$

where $\boldsymbol{D}$ is the matrix whose elements D_{kj} take the value 1 with probability p and 0 with probability $1 - p$. Observe that when $p = 0$ all the elements $D_{kj} = 0$; when this happens we

set $V_k(\boldsymbol{d}) = 0$. In this condition, the kth member possesses no knowledge about the fitness function, and his choices are driven only by consensus seeking. By increasing p from 0 to 1 we control the level of knowledge of the members, which affects the ability of the group in maximizing the fitness function equation (4.1). Note that the choice configuration that optimizes the perceived fitness equation (4.2), does not necessarily optimize the group fitness equation (4.1). This makes the mechanism of social interactions, by means of which knowledge is transferred, crucial for achieving high-performing decision-making process.

All members of the group make choices on each of the N decision variables d_j. Therefore, the state of the kth member ($k = 1, 2, \dots, M$) is identified by the N-dimensional vector $\sigma_k = (\sigma_k^1, \sigma_k^2, \dots, \sigma_k^N)$, where $\sigma_k^j = \pm 1$ is a binary variable representing the opinion of the kth member on the jth decision. For any given decision variable d_j, individuals k and h agree if $\sigma_k^j = \sigma_h^j$, otherwise they disagree. Within the framework of Ising's approach, (Sornette, 2014), (Ising, 1925), (Brush, 1967), disagreement is characterized by a certain level of conflict E_{kh}^j (energy level) between the two socially interacting members k and h, i.e. $E_{kh}^j = -\frac{J}{\langle k \rangle}\sigma_k^j \sigma_h^j$, where $J/\langle k \rangle$ is the strength of the social interaction and $\langle k \rangle$ the mean degree of the network of interactions between agents. Therefore, the total level of conflict on the decision d_j is given by:

$$E^j = -\sum_{(k,h)} \frac{J}{\langle k \rangle}\sigma_k^j \sigma_h^j \tag{4.3}$$

where the symbol () indicates that the sum is limited to the nearest neighbors, i.e. to those individuals which are directly connected by a social link. A multiplex network, (De Domenico et al., 2013), (Lee et al., 2015), (Boccaletti et al., 2014), (Wang et al., 2015), (Kivela et al, 2014), with N different layers is defined. On each layer, individuals share their opinions on a certain decision variable d_j leading to a certain level of conflict E^j. The graph of social network on the layer d_j is described in terms of the symmetric adjacency matrix $\boldsymbol{A}^j$ with elements A_{kh}^j. The interconnections between different layers represent the interactions among the opinions of the same individual k on the decision variables. Fig. 4.1 shows an example of a complete multiplex network, where the dashed lines, connecting the different decision layers, represent the interaction between the opinions that each member has on the decision variables. This interaction occurs via the *NK* perceived fitness, i.e. changing the opinion on the decision variable j causes a modification of the perceived pay-off, which also depends on the opinions the member has on the remaining decision variables.

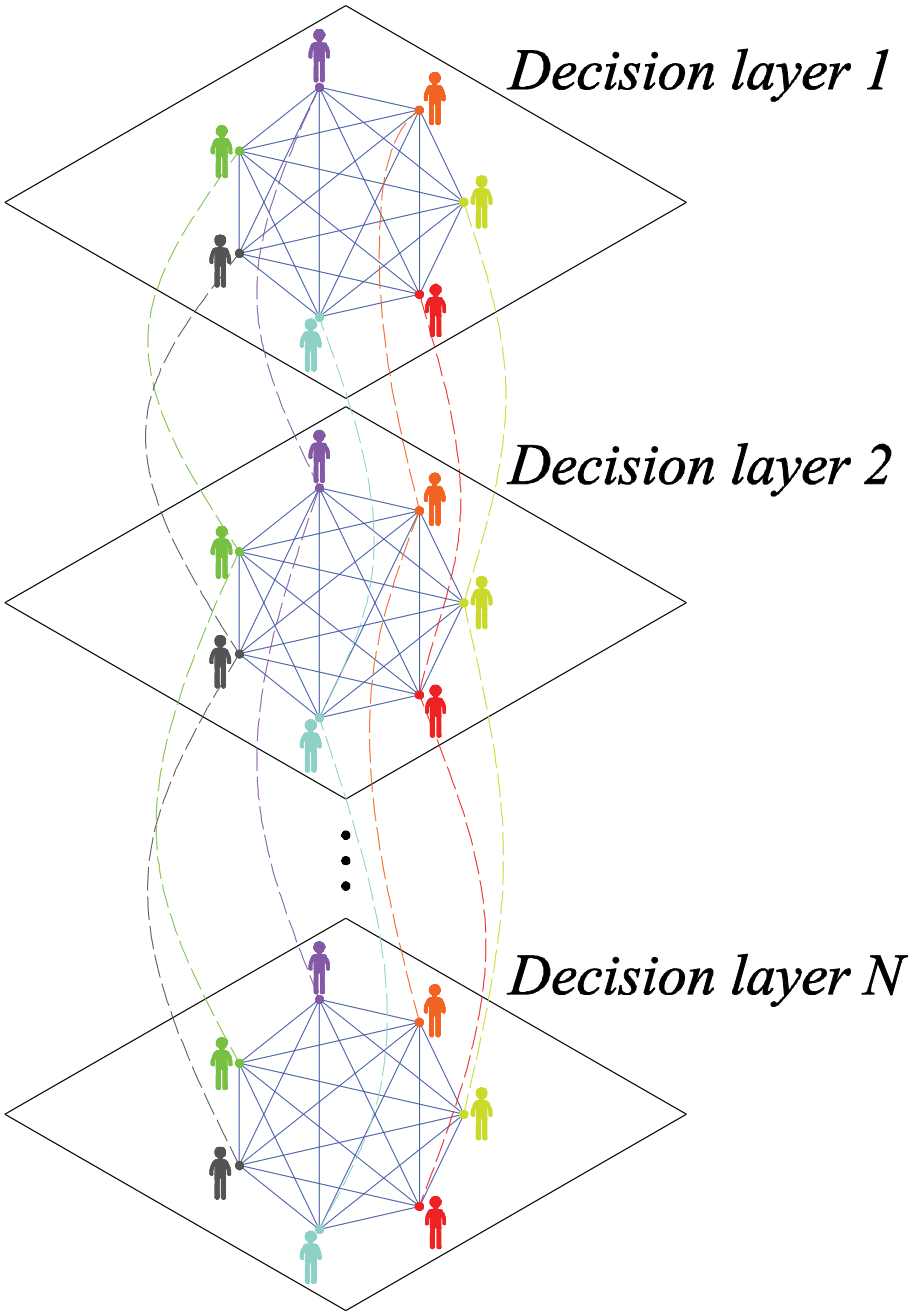


Fig. 4.1, An *N*-layer multiplex complete network. Each layer is associated with one single decision variable. Blue lines represent social links between members, whereas dashed lines represent the interconnections between the decisions of each member.

In order to model the dynamics of decision-making in terms of a continuous-time Markov process, we define the state vector $\boldsymbol{s}$ of size $n = M \times N$ as:

$$\boldsymbol{s} = (s_1, s_2, \dots, s_n) = (\sigma_1^1, \sigma_2^1, \dots, \sigma_M^1, \sigma_1^2, \sigma_2^2, \dots, \sigma_M^2, \dots, \sigma_1^N, \sigma_2^N, \dots, \sigma_M^N)$$

and the block diagonal adjacency matrix $\boldsymbol{A} = \mathrm{diag}(\boldsymbol{A}^1, \boldsymbol{A}^2, \dots, \boldsymbol{A}^N)$. For any given lth component $s_l = \sigma_k^j$ of the vector $\boldsymbol{s}$ it is possible to uniquely identify the member k and the

decision variable *j* by means of the relations $k = \text{quotient}(l-1, M) + 1$, and $j = \text{mod}(l-1, M) + 1$. The total level of conflict can be then rephrased as:

$$E(\boldsymbol{s}) = -\frac{1}{2}\frac{J}{\langle k \rangle}\boldsymbol{A}\boldsymbol{s}\cdot\boldsymbol{s} = -\frac{1}{2}\frac{J}{\langle k \rangle}\sum_{ij} A_{ij}s_i s_j \tag{4.4}$$

Observe that $A_{ii} = 0$ (with $i = 1, \dots, n$). In equation (4.4) the term 1/2 avoids that each couple of agents *k* and *h* be double counted. Now let be $P(\boldsymbol{s}, t)$ the probability that, at time *t*, the state vector takes the value $\boldsymbol{s}$ out of 2^n possible states. The time evolution of the probability $P(\boldsymbol{s}, t)$ obeys the master equation

$$\frac{dP(\boldsymbol{s}, t)}{dt} = -\sum_l w(\boldsymbol{s}_l \to \boldsymbol{s}'_l)P(\boldsymbol{s}_l, t) + \sum_l w(\boldsymbol{s}'_l \to \boldsymbol{s}_l)P(\boldsymbol{s}'_l, t) \tag{4.5}$$

where $\boldsymbol{s}_l = (s_1, s_2, \dots, s_l, \dots, s_n)$ and $\boldsymbol{s}'_l = (s_1, s_2, \dots, -s_l, \dots, s_n)$. The transition rate $w(\boldsymbol{s}_l \to \boldsymbol{s}'_l)$ of the Markov chain (i.e. the probability per unit time that the opinion s_l flips to $-s_l$ while the others remain temporarily fixed) is defined so as to be the product of the transition rate of the Ising-Glauber dynamics (Glauber, 1963), (see also Appendix 8.2), which models the process of consensus seeking to minimize the conflict level, and the Weidlich exponential rate $\exp\{\beta'[\Delta V(\boldsymbol{s}'_l, \boldsymbol{s}_l)]\}$ (Weidlich, 1991), (Schweitzer & Farmer, 2007), which models the self-interest behavior of the agents:

$$w(\boldsymbol{s}_l \to \boldsymbol{s}'_l) = \frac{1}{2}\left[1 - s_l \tanh\left(\beta\frac{J}{\langle k \rangle}\sum_h A_{lh}s_h\right)\right] \times \exp\{\beta'[\Delta V(\boldsymbol{s}'_l, \boldsymbol{s}_l)]\} \tag{4.6}$$

The quantity β is the inverse of the social temperature that is a measure of the degree of confidence the members have in the other judgment/opinion. Similarly, the quantity β' is related to the level of confidence the members have about their perceived fitness (the higher β', the higher the confidence). In equation (4.6) the pay-off function $\Delta V(\boldsymbol{s}'_l, \boldsymbol{s}_l) = \bar{V}(\boldsymbol{s}'_l) - \bar{V}(\boldsymbol{s}_l)$, where $\bar{V}(\boldsymbol{s}_l) = V_k(\sigma_k)$, is simply the change of the fitness perceived by the agent $k = \text{quotient}(l-1, M) + 1$, when its opinion $s_l = \sigma_k^j$ on the decision $j = \text{mod}(l-1, M) + 1$ changes from $s_l = \sigma_k^j$ to $s'_l = -\sigma_k^j$. Note that equation (4.6) satisfies the detailed balance condition (see Appendix 8.3).

To solve the Markov process equations (4.5) and (4.6), we employ a simplified version of the exact stochastic simulation algorithm proposed by Gillespie, (Gillespie, 1976-1977). More details about the algorithm are provided in Appendix 8.4. The algorithm allows to generate a statistically correct trajectory of the stochastic process equations (4.5) and (4.6).

4.2 Measuring the performance of the decision-making process

As the process evolves the bitstring $\boldsymbol{d} = (d_1, d_2, \dots, d_N)$ of the decisions of the group of agents needs to be determined at each time t. Different choices can be made. Among these, the majority rule seems appropriate, especially in presence of cognitive limits of the agents, as it does not need to query the value of fitness perceived by each agent. In this case, given the set of opinions $(\sigma_1^j, \sigma_2^j, \dots, \sigma_M^j)$ that the agents have about the decision j, at time t, we set:

$$d_j = \operatorname{sgn}\left(M^{-1}\sum_{k=1}^{M}\sigma_k^j\right), \quad j = 1, 2, \dots, N. \tag{4.7}$$

If M is even and in the case of a parity condition, d_j is, instead, uniformly chosen at random between the two possible values ± 1. The group fitness is then calculated as $V[\boldsymbol{d}(t)]$ and the ensemble average $\langle V(t)\rangle$ is then evaluated. When the landscape is not too large, i.e. $(N < 13)$, it is possible to estimate the efficacy of the group in optimizing $\langle V(t)\rangle$ in terms of normalized average fitness $\langle V(t)\rangle / V_{max}$ where $V_{max} = max[V(\boldsymbol{d})]$.

The consensus of the members on the decision variable *j* is measured as follows. We define the average opinion $\bar{\sigma}^j$ of the group on the decision *j*

$$\bar{\sigma}^j = \frac{1}{M}\sum_{k=1}^{M}\sigma_k^j \tag{4.8}$$

Note that the quantity $\bar{\sigma}^j$ ranges in the interval $-1 \leq \bar{\sigma}^j \leq 1$, and that $\bar{\sigma}^j = \pm 1$ only when full consensus is reached. Therefore, a possible measure of the consensus among the members on the decision variable *j* is given by the ensemble average of the time-dependent quantity $\chi^j = \left(\bar{\sigma}^j\right)^2 \in [0, 1]$, i.e.,

$$\chi^j(t) = \frac{1}{M^2}\sum_{hk}\langle\sigma_k^j(t)\sigma_h^j(t)\rangle = \frac{1}{M^2}\sum_{hk}R_{hk}^j(t) \tag{4.9}$$

Observe that $\langle\sigma_k^j(t)\sigma_h^j(t)\rangle = R_{hk}^j(t)$ is the correlation function of the opinions of the members *k* and *h* on the same decision variable *j*. Given this, a possible ansatz to measure the entire consensus of the group on the whole set of decisions is:

$$\chi(t) = \frac{1}{N}\sum_{j}\chi^j(t) = \frac{1}{M^2N}\sum_{j=1}^{N}\sum_{hk=1}^{M}R_{hk}^j(t) \tag{4.10}$$

Note that $0 \leq \chi(t) \leq 1$.

4.3 Simulations and Results

We consider initially a group of $M = 6$ members which have to make $N = 12$ decisions. For the sake of simplicity, the network of social interactions on each decision layer j is described by a complete graph, where each member is connected to all the others. We also set $\beta' = 10$, since we assume that the agents have a good confidence about their perceived fitness.

We simulate many diverse scenarios to investigate the influence of the parameter p, i.e. of the level of knowledge of the members, and the effect of the parameter βJ on the outcome of the decision-making process. The simulation is stopped at steady-state. This condition is identified by simply taking the time-average of consensus and pay-off over consecutive time intervals of fixed length T and by checking that the difference between two consecutive averages is sufficiently small.

For any given p and βJ, each stochastic process equations (4.5) and (4.6) is simulated by generating *100* different realizations (trajectories). For each single realization, the competence matrix $\boldsymbol{D}$ is set, and the initial state of the system is obtained by drawing from a uniform probability distribution, afterward the time evolution of the state vector is calculated with the stochastic simulation algorithm [see Appendix 8.4].

Fig. 4.2 shows the time-evolution of normalized average fitness $\langle V(t)\rangle/V_{max}$, for $p = 0.5$ (i.e. for a moderate level of knowledge of the members), different values of the complexity parameter $K = 1, 5, 11$, and different values of $\beta J = 0.0, 2.5, 5.0$.

We observe that for $\beta J = 0.0$, i.e. in absence of social interactions [Fig. 4.2 (a)], the decision-making process is strongly inefficient, as witnessed by the very low value of the average fitness of the group. Every one of the group makes his/her choices to optimize the perceived fitness, but, because of the absence of social interactions, he/she behaves independently from the others and does not receive any feedback about the actions of the other group members. Hence, individuals remain close to their local optima, group fitness cannot be optimized [Fig. 4.2 (a)], and the consensus is low [Fig. 4.2 (b)].

As the strength of social interactions increases, i.e., $\beta J = 2.5$ [Fig. 4.2 (c)], members can exchange information about their choices. Social interactions push the individuals to seek consensus with the member who is experiencing higher payoff. In fact, on the average, those members, which find a higher increase of their perceived fitness, change opinion much faster than the others. Thus, the other members, while seeking consensus, skip the local optima of

their perceived fitness and keep exploring the landscape, leading to a substantial increase of the group performance both in terms of group fitness values [Fig. 4.2 (c)], as well as in terms of final consensus [Fig. 4.2 (d)]. Thus, the system collectively shows a higher level of knowledge and higher ability in making good choices than the single members (i.e., a higher degree of intelligence).

It is noteworthy that when the strength of social interactions is too large, $\beta J = 5.0$ [Fig. 4.2 (e)], the performance of the group in terms of fitness value worsens. In fact, very high values of βJ, accelerating the achievement of consensus among the members [Fig. 4.2 (f)], significantly impede the exploration of the fitness landscape and hamper that change of opinions can be guided by payoff improvements. The search of the optimum on the fitness landscape is slowed down, and the performance of the collective decision-making decreases both in terms of the time required to reach the steady-state as well as in terms of group fitness.

Fig. 4.2 shows that rising the complexity of the landscape, i.e. increasing K, negatively affects the performance of the collective decision-making process, but does not qualitatively change the behavior of the system. However, figure 4.2 (c) also shows that, to cause a significant worsening of the group fitness, K must take very large values, i.e., $K = 11$. Instead, at moderate, but still significant, values of complexity (see results for $K = 5$) the decision-making process is still very effective, leading to final group fitness values comparable to those obtained at the lowest level of complexity, i.e., at $K = 1$.

In Figures 4.2 (b, d, f) the ensemble average $\chi(t)$ of the consensus among the members is shown as a function of time t, for $p = 0.5, K = 1, 5, 11$, and for different values of $\beta J = 0.0, 2.5, 5.0$.

At $\beta J = 0$, the consensus is low. In this case, at each time *t*, members' opinions are random variables almost uniformly distributed between the two states ± 1. Hence, the quantity $\chi(t)$ can be analytically calculated as $\chi(t) \approx 1/M$. For $M = 6$ this gives $\chi(t) \approx 0.16$, which is just the average value observed in figure 4.2 (b).

As the strength of social interactions rises, members more easily converge toward a common opinion. However, the random nature of the opinion dynamics still prevents full agreement from being achieved, see figure 4.2 (d). This, as observed in figure 4.2 (a), has a very beneficial effect as individuals continue exploring the fitness landscape looking for maxima, thus leading to higher performance of the collective decision-making process.

However, when the strength of social interactions is significantly increased, a very high value of consensus among members is rapidly achieved [Fig. 4.2 (f)], the exploration of the landscape is slowed down, and the performance of the decision making-process significantly worsen [Fig. 4.2 (e)].

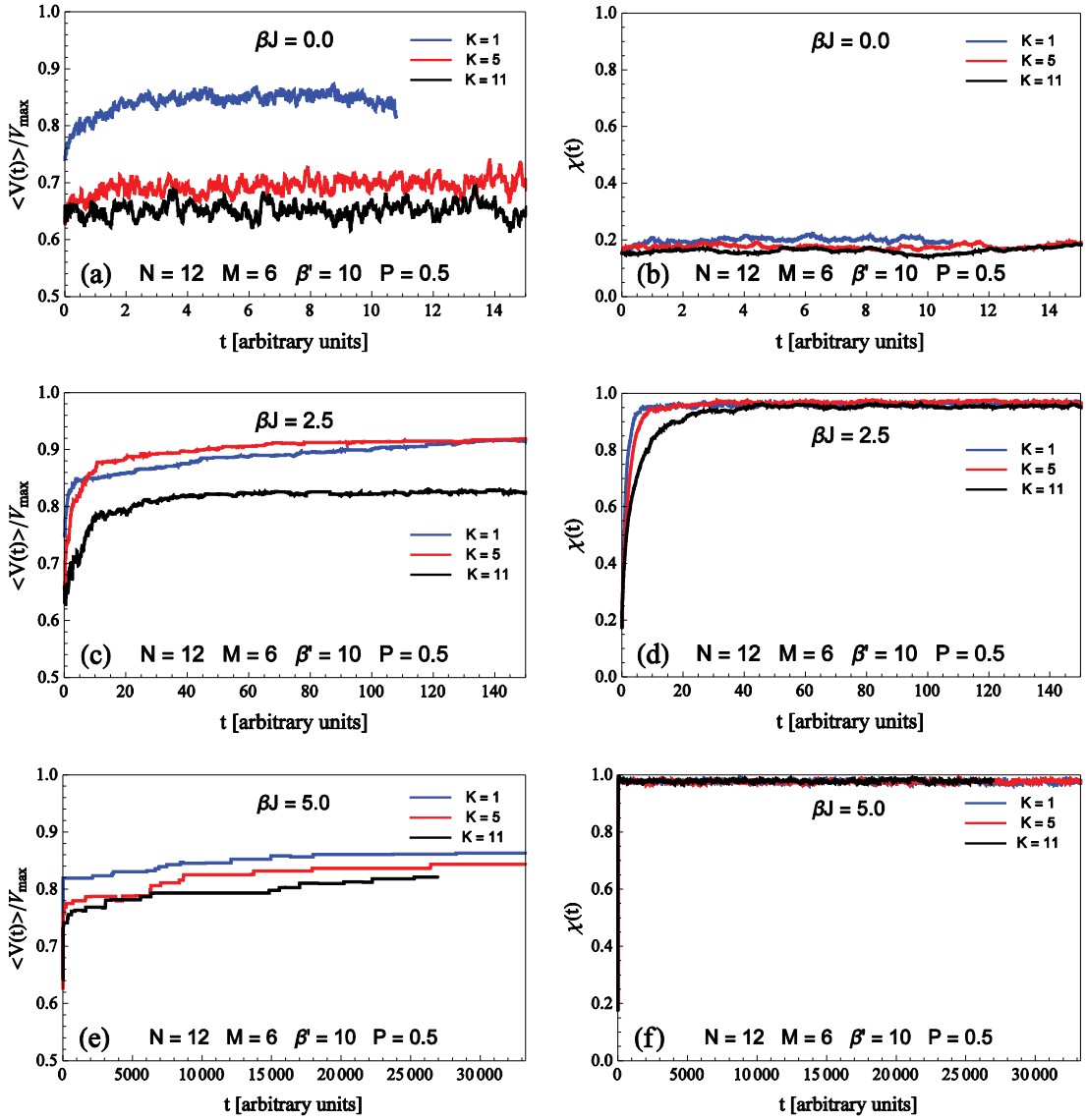


Fig. 4.2, The time-evolution of the normalized average group fitness $\langle V(t)\rangle/V_{max}$ and statistically averaged consensus $\chi(t)$, for p = 0.5, K = 1, 5, 11 and (a-b) βJ = 0.0, (c-d) βJ = 2.5, (e-f) βJ = 5.0.

We, then, expect that, given β' and K, an optimum of βJ exists, which maximizes the steady-state fitness of the group. This is, indeed, confirmed by the analysis shown in Figure 4.3, where the steady-state values of the normalized group fitness $\langle V_\infty\rangle/V_{max} = \langle V(\mathrm{t} \to \infty)\rangle/V_{max}$ [Fig. 4.3 (a)], and social consensus $\chi_\infty = \chi(\mathrm{t} \to \infty)$ [Fig. 4.3 (b)] are plotted as a function of βJ, for $p = 0.5$ and the three considered values of $K = 1, 5, 11$.

Results in Figure 4.3 (a) stresses that the fitness landscape complexity (i.e., the parameter K) marginally affects the performance of the decision-making process in terms of group fitness, provided that K does not take too high values. In fact, curves calculated for $K = 1, 5$ run close to each-other.

More interesting, Fig. 4.3 shows that increasing βJ from zero, makes both $\langle V_\infty \rangle / V_{max}$ and χ_∞ rapidly increase. This increment is, then, followed by a region of a slow change of $\langle V_\infty \rangle / V_{max}$ and χ_∞. It is worth noticing, that the highest group fitness value is obtained at the boundary between the increasing and almost stationary regions of χ_∞.

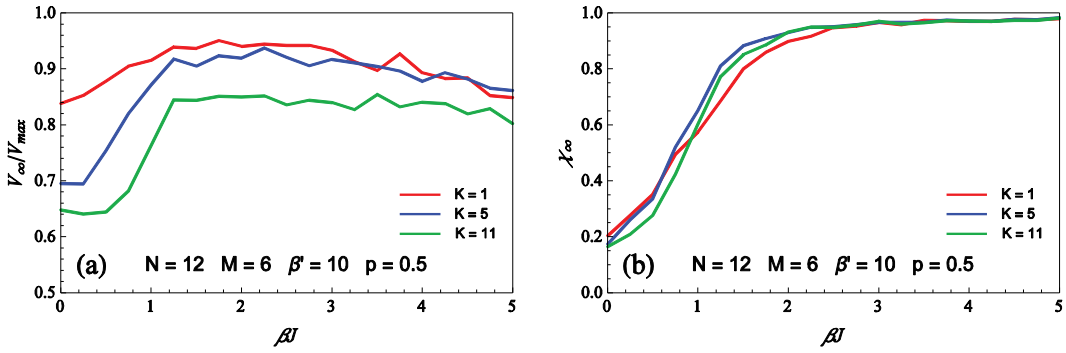


Fig. 4.3, The stationary values of the normalized averaged fitness $\langle V(t) \rangle / V_{max}$ and statistically averaged consensus $\chi(t)$ as a function of βJ. Results are presented for p = 0.5, K = 1, 5, 11.

In Figure 4.4 we investigate the influence of the level of knowledge p of members on the time-evolution of the normalized average fitness $\langle V_\infty \rangle / V_{max}$. Results are presented for $\beta J = 2.5, K = 1, 5, 11$, and for different values of $p = 0.05, 0.1, 0.2, 0.6, 1.0$.

Results show that improving the knowledge of the members, i.e. increasing p, enhances the performance of the decision-making process. A higher steady-state normalized fitness $\langle V_\infty \rangle / V_{max}$, and a faster convergence toward the steady state are observed. Note also, that, especially in the case of high complexity, [Figs. 4.4 (e, f)], increasing p above *0.2* reduces the fluctuations of $\langle V(t) \rangle$, because of the higher agreement achieved among the members at the higher level of knowledge.

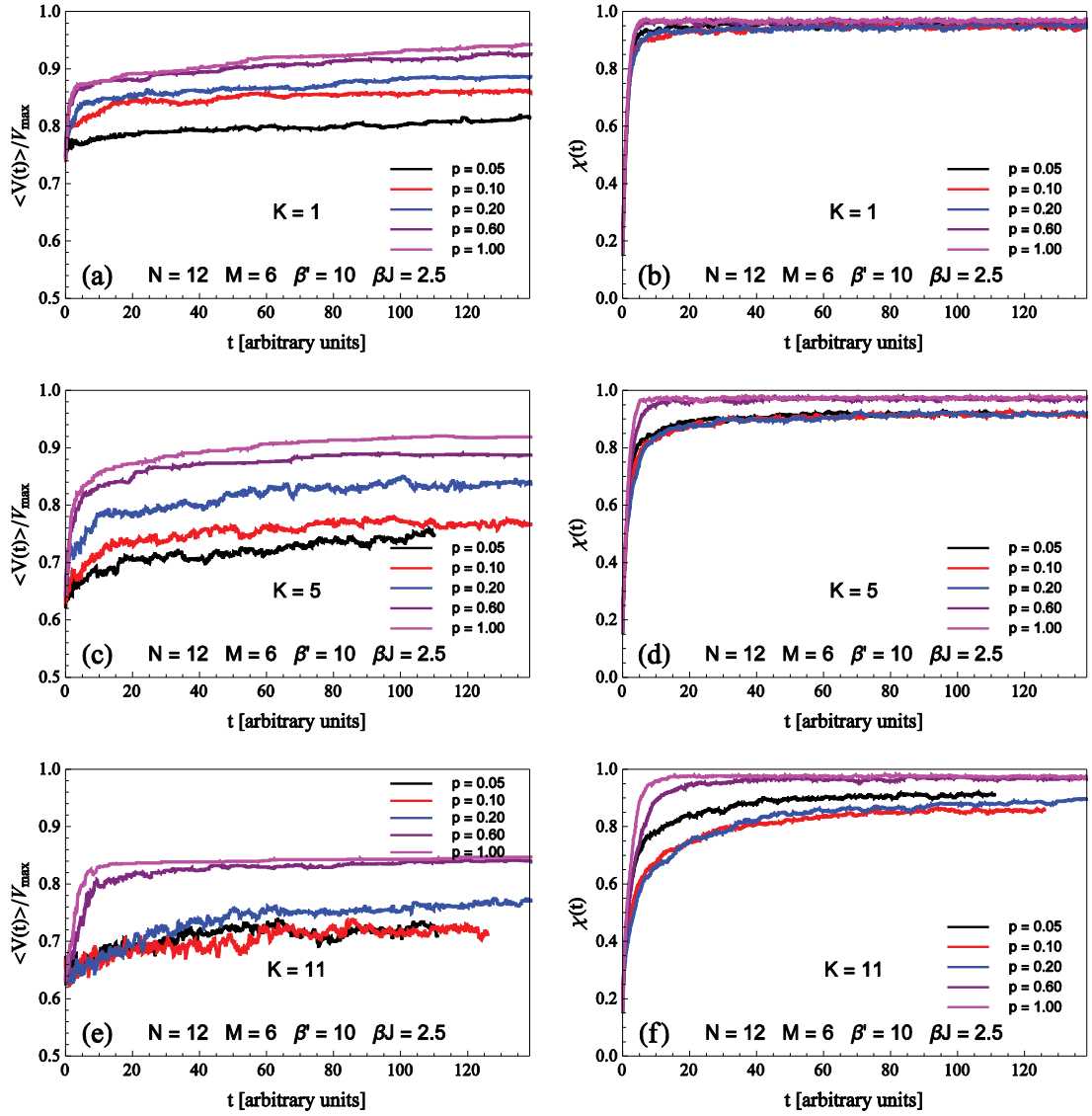


Fig. 4.4, The time-evolution of the normalized average group fitness $\langle V(t)\rangle/V_{max}$ and statistically averaged consensus $\chi(t)$, for $\beta J = 2.5$, $p = 0.05$, 0.1, 0.2, 0.6, 1 and (a, b) K =1, (c, d) $K = 5$ and (e, f) $K = 11$.

In Figure 4.5 the steady-state values of the normalized group fitness $\langle V_\infty\rangle/V_{max}$ [Fig. 4.5 (a)], and social consensus χ_∞ [Fig. 4.5 (b)] are shown as a function of p, for $\beta J = 2.5$ and the three considered values of $K = 1, 5, 11$. Note that as p is increased from zero, the steady state value $\langle V_\infty\rangle/V_{max}$ initially grows fast [Fig. 4.5 (a)]. In fact, because of social interactions, increasing the knowledge of each member also increases the knowledge of the group as a whole. But, above a certain threshold of p, the increase of $\langle V_\infty\rangle/V_{max}$ is much less significant. This indicates that the knowledge of the group is subjected to a saturation effect. Therefore, a moderate level of knowledge is already enough to guarantee very good performance of

decision-making process; higher knowledge levels being only needed to accelerate the convergence of the decision-making process.

Fig. 4.5 (b) shows that for vanishing values of p the consensus χ_∞ takes high values, as each member's choice is driven only by consensus seeking. Increasing p initially causes a decrease of consensus, as the self-interest of each member leads to a certain level of disagreement. However, a further increment of p makes the members' knowledge overlap so that the self-interest of each member almost points in the same direction, resulting in a consensus increase.

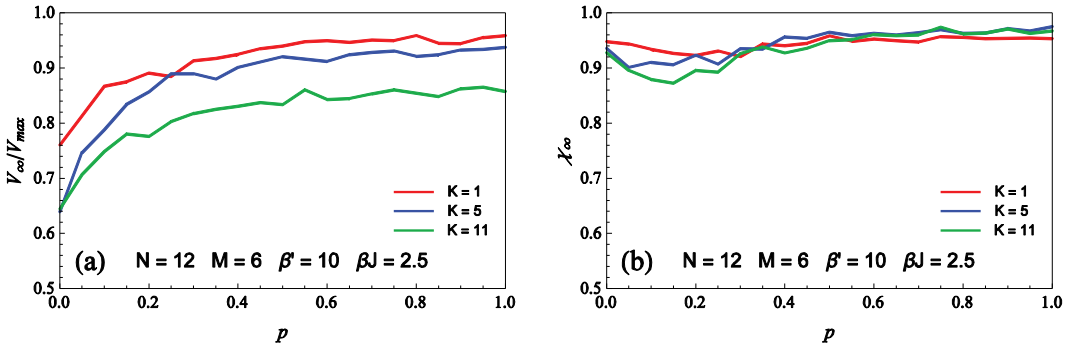


Fig. 4.5, The stationary values of the normalized average group fitness $\langle V(t)\rangle/V_{max}$ and statistically averaged consensus $\chi(t)$, as a function of p. Results are presented for $\beta J = 2.5$, $\beta' = 10$ and $K = 1, 5, 11$.

4.4 Critical transition and collective intelligence

The previously shown results point out that high consensus is necessary to guarantee high efficacy of the decision-making process, i.e. high values of $\langle V_\infty\rangle/V_{max}$. This suggests that the decision-making becomes optimal, i.e. the group as a whole is characterized by a higher degree of intelligence, at the point where the system dynamics changes qualitatively. This aspect of the problem is investigated in Figure 4.6, where the stationary values of efficacy $V_\infty = \langle V(t \to \infty)\rangle$ and the degree of consensus $\chi_\infty = \langle \chi(t \to \infty)\rangle$ are reported as a function of the quantity βJ for different group sizes $M = 6, 12, 24$, $N = 12$, $K = 5$, $\beta' = 10$ and for an average level of knowledge $p = 0.5$. In this case, 100 different realizations of the same process have been simulated.

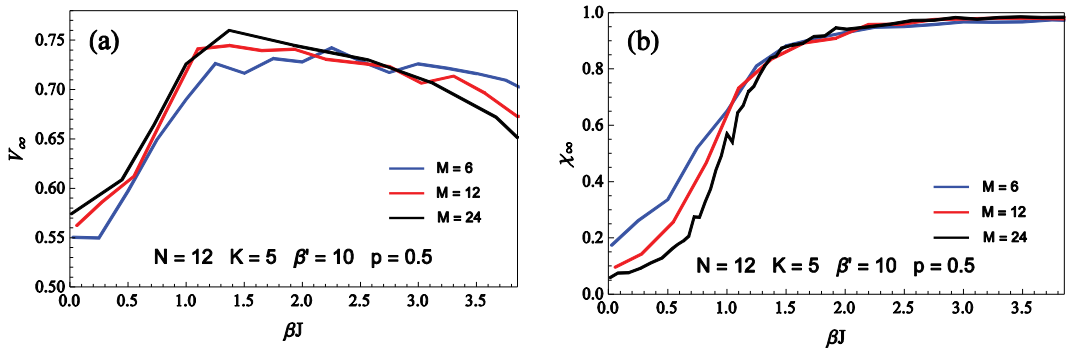


Fig. 4.6, The stationary values of the averaged fitness V_∞ and statistically averaged consensus $\chi(t)$ as a function of βJ. Results are presented for $N = 12$, $K = 5$, $p = 0.5$, $\beta' = 10$, and for three different team sizes: $M = 6, 12, 24$.

Results clearly show that a critical threshold value of βJ exists at which both consensus and payoff have a sharp and concurrent increase. Notably, the transition from low to high payoff, accompanied by an analogous transition from low to high consensus, becomes sharper as the group size M is increased. However, in all cases, given $\beta' = 10$, the transition occurs for $(\beta J)_c \approx 1$. Interestingly this threshold value is close to the critical ordering transition of the Ising model on a complete graph, in the thermodynamic limit of large M. This result can be obtained by using the findings by Vespignani and Mendes (Leone, Vázquez, Vespignani, & Zecchina, 2002), (Dorogovtsev, Goltsev, & Mendes, 2002), who independently demonstrated that for general graphs the critical transition of the Ising model occurs at

$$\left(\beta \frac{J}{\langle k \rangle}\right)_c = -\frac{1}{2}\log\left(1 - 2\frac{\langle k \rangle}{\langle k^2 \rangle}\right) \tag{4.11}$$

Thus, considering that for a complete graph $\langle k \rangle = M - 1$, $\langle k^2 \rangle = (M-1)^2$ and that M is large, expanding Eq. (4.11) at first order in $\langle k \rangle / \langle k^2 \rangle$ gives $(\beta J)_c = 1$.

However, we expect that the threshold $(\beta J)_c$ is affected by the values of β'. Calculations confirm this trend as shown in Fig. 4.7, where the response surfaces in terms of the steady state fitness V_∞ and the consensus χ_∞ are plotted as a function of βJ and β' for $N = 15$, $K = 14$, $p = 1$ and for two team size, $M = 7$ and $M = 21$. Given the input parameters, βJ and β', calculations have been carried out by simulating 500 different realizations and then taking the average. The presence of a critical front (line) is clearly observed. It is identified by the values of βJ and β' at which a concurrent transition from low value to high (or vice versa from high to low) values of both consensus and group fitness takes place.

Interestingly there is a relatively small region Ω, very close to the transition front, where the performances of the group are maximized.

Comparing the surfaces [Fig. 4.7 (a, c)], about the team size $M = 7$, with the ones about $M = 21$ [Fig. 4.7 (b, d)], we observe that the overall trend is almost the same, but the region of high performance is slightly larger for $M = 7$ compared to $M = 21$, and the critical transition less sharp. This explains why in the "social" literature it is well known that bigger groups perform worse than small groups in making decisions. Moreover, the performance peak is approximatively the same, possibly because $M = 7$ is an already enough group size for accurately solve a $N = 15$ $K = 14$ problem in complete knowledge $p = 1$. Probably on more complex landscapes and uncomplete knowledge $p < 1$, it could result appropriate to increase the number of agents to get better performance.

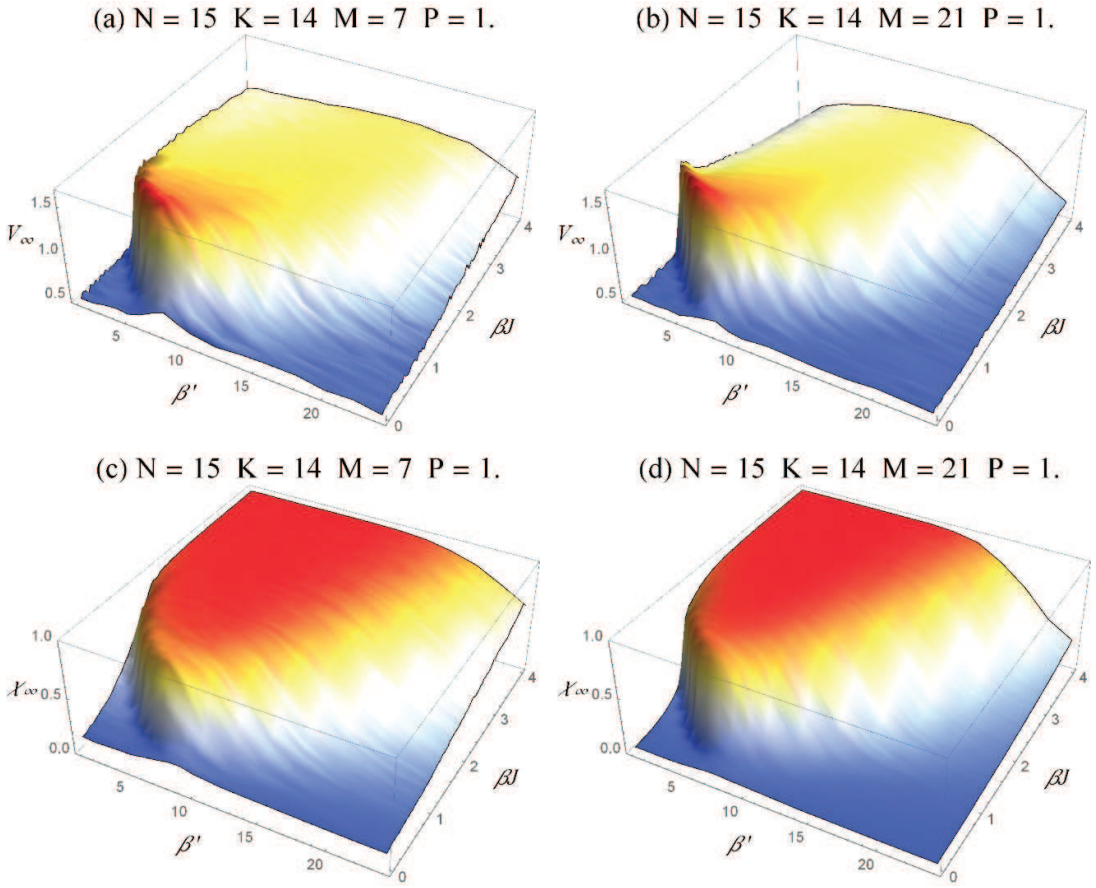


Fig. 4.7, Response surfaces in terms of the steady state fitness V_∞ and the consensus χ_∞, as a function of βJ and β', for $N = 15$, $K = 14$, $p = 1$ and $M = 7$ and $M = 21$.

In all cases, the collective intelligence of the group (i.e. the ability to make decisions leading to high values of the group fitness) sets in only if also the consensus sets in, and this happens exactly at the critical front. To support this conclusion, we calculate the mutual information between the two stochastic variables: the group fitness V_∞ and the consensus

χ_∞. The mutual information $MI(x,y)$ between two continuously distributed stochastic variables x and y is

$$MI(x,y) = \int dx\, dy\, p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} \tag{4.12}$$

This quantity (see Appendix 8.6 for more details) is a measure of the information gained about the behavior of one random variable, say x, by observing the behavior of the other variable y. Hence, the mutual information measures the difference between the initial uncertainty on the variable x and the uncertainty that remains about x after the observation of the behavior of the variable y. Under this perspective, it measures the information flow from the variable y to the variable x. Also, observe that $MI(x,y) = MI(y,x)$.

Fig. 4.8 shows that the quantity $MI(\chi_\infty, V_\infty)$ is maximized just at the critical transition, i.e. on the critical front the information flow from the complex fitness landscape to the group of agents is strongly enhanced. This allows to identify solutions with much higher group fitness. In other words, we may say that, at the critical threshold, the exchange of information promoted by social interactions, provides the group with higher collective knowledge about the group fitness landscape. This leads to an improved exploration of the landscape, to better choices and finally to the emergence of the ″collective intelligence″ of the group.

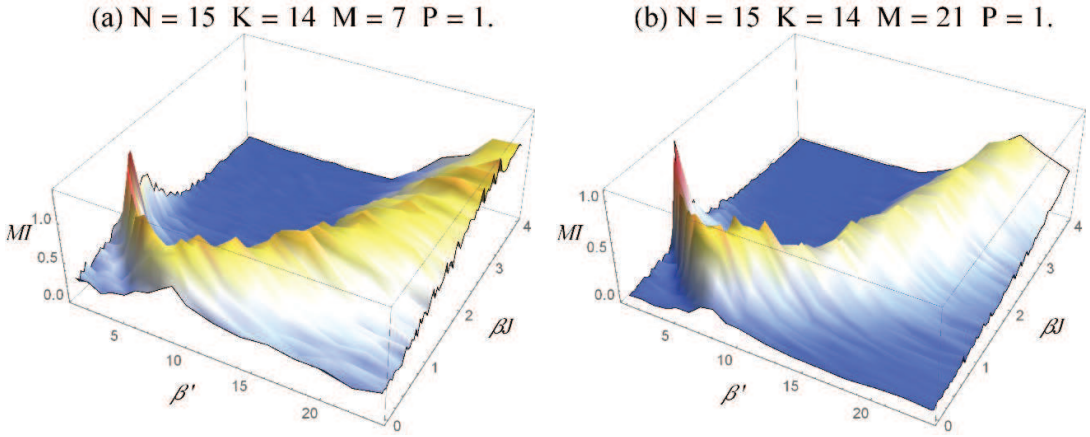


Fig. 4.8, Mutual information surface $MI(\chi_\infty, V_\infty)$ between the group fitness V_∞ and the consensus χ_∞, as a function of βJ and β', for $N = 15$, $K = 14$, $p = 1$, (a) $M = 7$ and (b) $M = 21$.

Figures 4.9 and 4.10 show the response surfaces, analogues to ones reported in figures 4.7 and 4.8, in the case of limited knowledge of the agents, in particular for $p = 0.3$ [Fig. 4.9] and $p = 0.7$ [Fig. 4.10]. The trends are similar, with evident lower performance in terms of fitness values in the case of extremely low knowledge ($p = 0.3$) [Fig. 4.9], and instead almost indistinguishable in the case of medium-good knowledge ($p = 0.7$) [Fig. 4.10], compared to the complete knowledge case [Fig. 4.7]. At the critical threshold, the agents

with limited knowledge, driven by the social interactions, will follow, in making choices, those agents with higher knowledge, thus finally agreeing with them on the "good" choice.

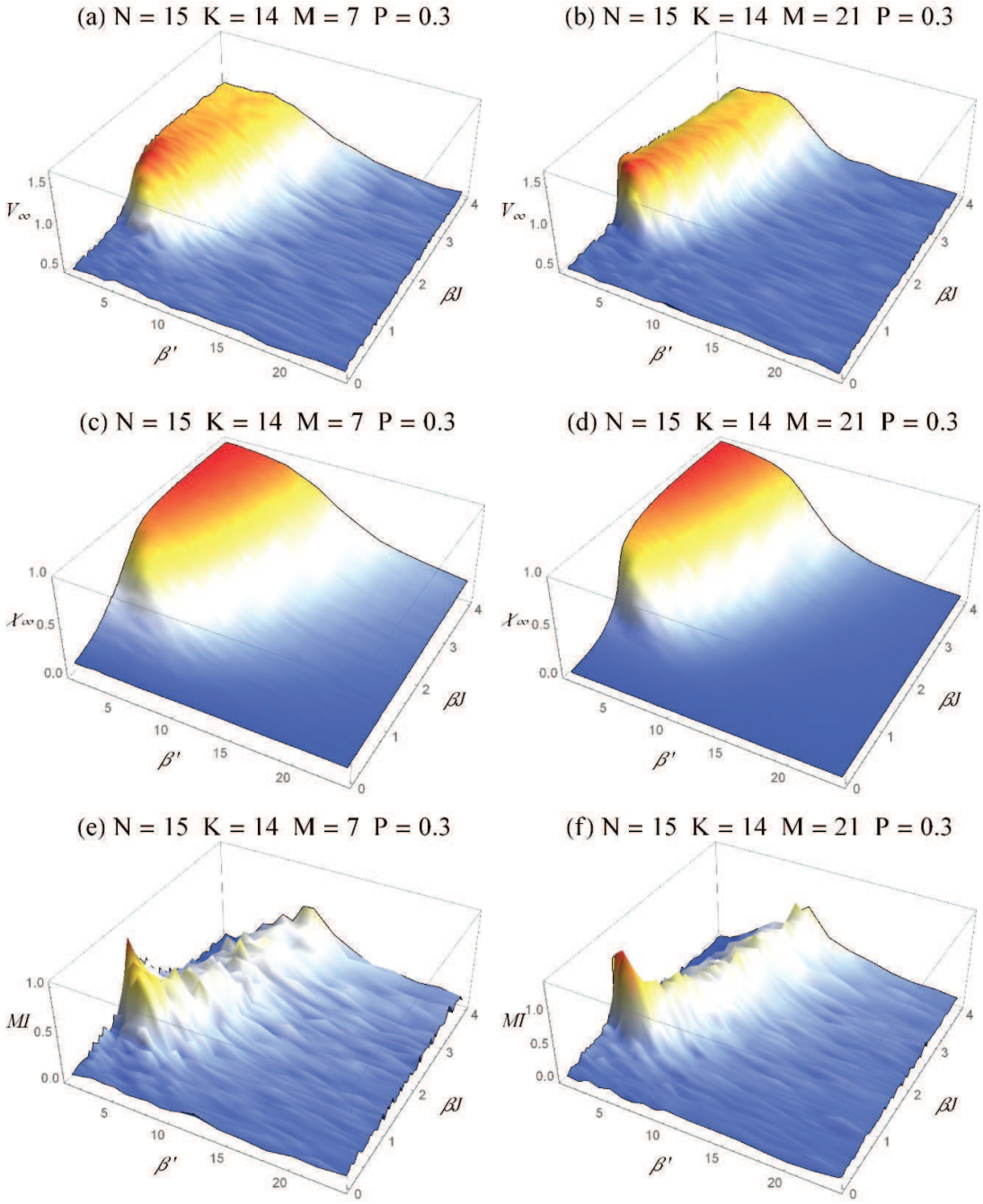


Fig. 4.9, Response surfaces in terms of (a-b) the steady state fitness V_∞, (c-d) consensus χ_∞ and (e-f) mutual information $MI(\chi_\infty, V_\infty)$, as a function of βJ and β', for $N = 15$, $K = 14$, $p = 0.3$ and $M = 7$ and $M = 21$.

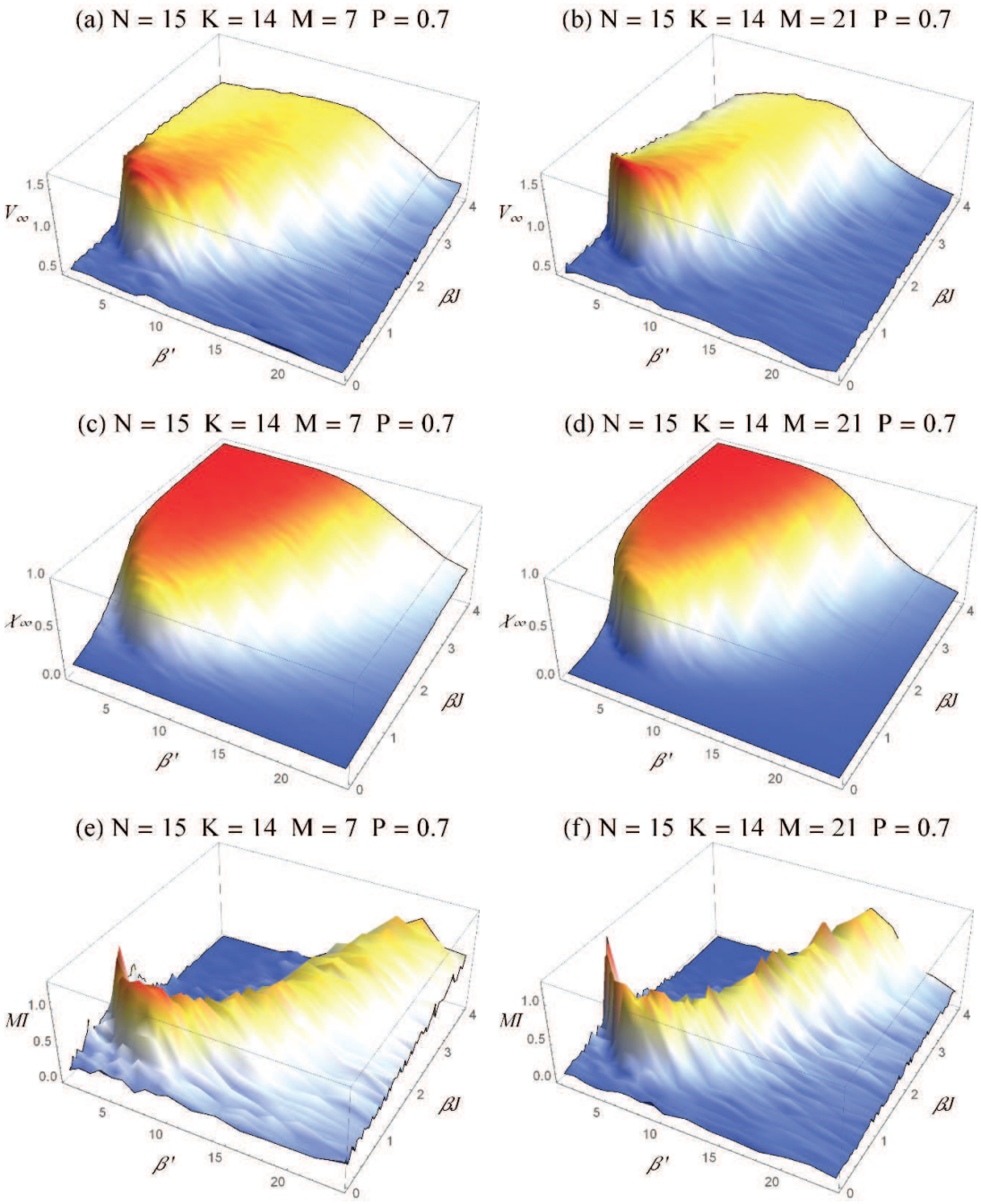


Fig. 4.10, Response surfaces in terms of (a-b) the steady state fitness V_∞, (c-d) consensus χ_∞ and (e-f) mutual information $MI(\chi_\infty, V_\infty)$, as a function of βJ and β', for $N = 15$, $K = 14$, $p = 0.7$ and $M = 7$ and $M = 21$.

4.5 Conclusions

In this chapter, the author introduces a new model of collective decision-making (DMM). The model described the time evolution of group choices in terms of a time-continuous Markov process, where the transition rates have been defined to capture the effect of the two main forces, which drive the change of opinion of the members of the group. These forces are the rational behavior which pushes each member to increase his/her self-interest, and the social interactions, which push the members to reach a common opinion.

Our study provides a contribution to the literature identifying under which circumstances collective decision making is more performing. We found that a moderate strength of social interactions allows for knowledge transfer among the members, leading to a higher knowledge level of the group as a whole. This mechanism, coupled with the ability to explore the fitness landscape, strongly improves the performance of the decision-making process. We identified that the threshold value of the social interaction strength, at which the entire group is characterized by the highest degree of collective intelligence, is just the critical threshold at which the flow of information from the fitness landscape to the group of agents is maximized, thus improving the abilities of the group to explore the fitness landscape searching for the optimal solution.

We also noticed that increasing the level of knowledge of the members improves performance. However, above a certain threshold the knowledge of the group saturates, i.e. the performance of the collective decision-making process becomes much less sensitive to the level of knowledge of each single member. Therefore, we can state that the collective decision-making is very high-performing already at a moderate level of knowledge of the members and that very high knowledge of all members only serves to accelerate the convergence of the decision-making process.

5 The effect of social network structure on team performance

"The strength of the team
is each individual member.
The strength of each member
is the team."
Phil Jackson

In today's highly complex and uncertain environment, teams are effective coordination mechanisms contributing to firm competitive advantage (Jackson, DeNisi & Hitt, 2003), (Kozlowski & Bell, 2003), (Lawler, Mohrman & Ledford, 1995), (Lynn, Reilly, & Akgün, 2000). Indeed, teams are more and more adopted by the organizations to address a very large variety of projects, ranging from new product development, R&D activities, production and marketing issues, and so on. For this reason, a wide body of literature has analyzed the determinants of team performance with the aim of providing important lessons on how to design highly effective teams (Kozlowski & Bell, 2003), (Lepine et al., 2008), (Sanna & Parks, 1997).

A recent stream of research has focused the attention on how *social network features* relate to team performance. Teams are viewed as social communities involving team members and members external to the team, such as those belonging to other teams in the organization. Nodes (members) are linked one with each other by social relationships (ties). Structural features of the team social networks such as density and connectivity have been shown to be associated with team performance (Balkundi & Harrison, 2006), (Chung & Jackson, 2013), (Newman, 2010), (Stauffer, 2008).

The relationship between the team organizational features and team performance has been also investigated in a parallel stream of research (Clark & Wheelwright, 1992), (Doolen Hacker & Van Aken, 2003), (Lawer et al., 1995), (Pelled et al, 1999), which has put in evidence that the team *organizational structure* is one of the most important drivers of team performance (Tatikonda & Montoya-Weiss, 2001).

However, so far these two lines of inquiry have remained independent. No study has captured the existence of an interaction between the team social network features and team organizational structure in affecting team performance.

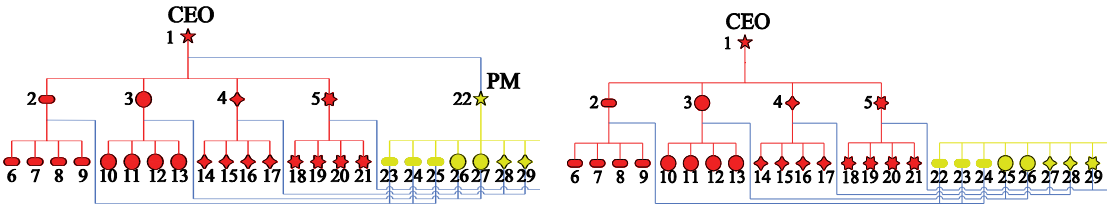
A relatively recent approach to study teams employs *social network theory* (Borgatti & Foster, 2003). Teams are framed as clusters of nodes (individuals), joined by a variety of links (relationships). As to the node, internal and external team networks are commonly distinguished. Internal networks are those involving only individuals belonging to the team, whilst external networks involve links between team members and external individuals, such as the members of others teams or the members of the organizations (Chung & Jackson, 2013). Social influence theory argues that social interactions are conduits of opinion formation, which stimulate convergence towards a common understanding of a situation and shared mental models among individuals. During the decision-making process, the team members receive two types of influence, because of their involvement in the hierarchical and social networks. The hierarchical ties exert pressure on team members because individuals usually prefer to avoid contrasts with the members in higher hierarchical positions.

Team members have an initial opinion on the decision to make, but then change it, because of their search for reaching consensus with interacting individuals (Horwitz & Horwitz, 2007), and at the same time trying to maximize their perceived payoff.

In this chapter, the effect of the strength of interactions, the team hierarchical structure and the topology of the social networks on team performance are investigated. In doing this, the decision-making model (DMM) previously proposed has been used with proper changes.

5.1 The computational model adapted to the case of study

We consider a team made by M agents (yellow), headed (*hierarchical* team) [Fig. 5.1 (a)] or not (*flat* team) [Fig. 5.1 (b)] by a Project Manager (PM), in charge to develop a product with N features. Each team member reports to the functional managers (nodes 2, 3, 4, 5 in fig. 5.2) of the organization (red) from which he/she comes from (same symbol). The organization is constituted by M_O agents, led by a chief executive officer (CEO), and divided into a certain number of departments (different symbols), each one with a fixed number of hierarchical levels and links per node. Each department, supervised by its functional manager, has a specialization, i.e. design, production, marketing, etc.

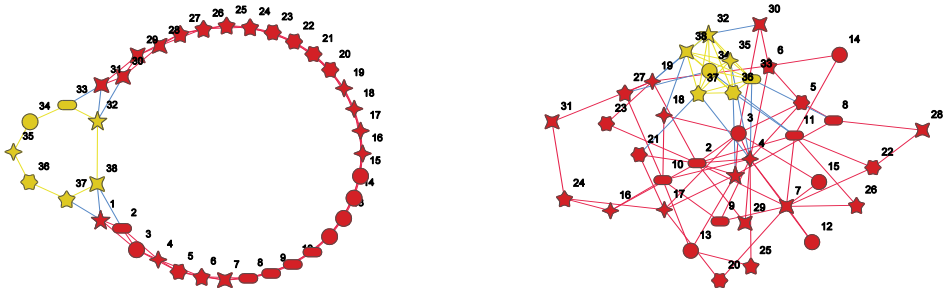


(a) (b)

Fig. 5.1, Examples of hierarchical networks: team members are in yellow color, organization members are in red. Blue links identify those members interacting through the external networks. (a) Hierarchical team, presence of the project manager (PM), (b) flat team, absence of the PM.

The individual opinions about the N decisions are modeled as binary variables, where +1 and –1 are the two possible values that each product attribute can assume. Examples of product attributes can be the product size (small or big), the target market (young or classic), the quality of materials (low or high), the production process (retail or series), and so on. The goal is to find, among the possible, the best set of decisions (product attributes), i.e. the one that brings to the highest fitness, that we can imagine directly linked to the gain of the organization.

Social interactions involving team members occur in two networks. The *hierarchical* network considers the hierarchical links, whereas the *social* network involves informal ties. The *global* network of interaction is the union of them. Between the different kinds of social network topology, we concentrated our attention on *small-world* and *scale-free* connectivity. Figure 5.2 reports an example of both the topologies. The main feature of small-world social networks [Fig 5.2 (a)], especially for very low density and rewiring probability values, is that they are similar to cyclic networks, with very few far connections. On the other hand, scale-free networks [fig. 5.2 (b)] with low-density values are characterized by very few nodes (agents) highly connected (hubs of the network) and most of the others poorly connected.



(a) (b)

Fig. 5.2, Examples of social networks with (a) small-world and (b) scale-free connectivity. Team members are in yellow color, organization members are in red color. Blue links identify those members interacting through the external network.

The fitness function is built employing the *NK* model, (Kauffman & Levin, 1987), (Kauffman S. A., 1989), (Weinberger, 1996). A N-dimensional vector space of decisions is considered, where each choice configuration is represented by a vector $\boldsymbol{d} = (d_1, d_2, \dots, d_N)$. As we already mentioned, each decision is a binary variable that may take only two values +1 or −1, i.e. $d_i = \pm 1, i = 1, 2, \dots, N$. The total number of decision vectors is, therefore, 2^N. Each vector $\boldsymbol{d}$ is associated with a certain fitness value $V(\boldsymbol{d})$ computed as the weighted sum of N stochastic contributions $W_j(\boldsymbol{d}_j^K)$, each decision leads to total fitness depending on the value of the decision d_j itself and the values of other K decisions d_i^j, $i = 1, 2, \dots, K$. Following the classical *NK* procedure, (more details are provided in Appendix 8.1), the quantities $W_j \in [0, 1]$ are determined as randomly generated 2^{K+1}-element "interaction tables". The fitness function of the group is defined as:

$$V(\boldsymbol{d}) = \frac{1}{N}\sum_{j=1}^{N} W_j(\boldsymbol{d}_j^K) \tag{5.1}$$

The integer index $K = 0, 1, 2, \dots, N-1$ is the number of interacting decision variables and tunes the complexity of the problem. The complexity of the problem increases with K.

Each agent has a different level of knowledge, i.e. knows the impact of a limited number of decisions on the overall payoff. The $(M + M_O) \times N$ competence matrix $\boldsymbol{D} = D_{kj}$ maps the knowledge of each agent k. $D_{kj} = 1$ if the member k knows the contribution of the decision j to the total fitness V, otherwise $D_{kj} = 0$.

We assume that each department of the organization is specialized on a certain number of decisions, i.e. the members of the organization have a *specialist knowledge*: they perceive only the contribution to the overall payoff of the decisions concerning the department which they belong. Differently, team members have a *general knowledge*: they have expertise about the decisions concerning the department from which they come from, and varying over the others with a probability p. For these decisions, we randomly choose $D_{kj} = 1$ with probability $p \in [0,\ 1]$ and $D_{kj} = 0$ with probability $1 - p$. By increasing p from 0 to 1, we control the level of knowledge of the team members.

Based on the level of knowledge each member k computes his/her own perceived fitness (self-interest) as:

$$V_k(\boldsymbol{d}) = \frac{\sum_{j=1}^{N} D_{kj} W_j(\boldsymbol{d}_j^K)}{\sum_{j=1}^{N} D_{kj}} \tag{5.2}$$

All agents make choices on each of the N decision variables d_j. Therefore, the state of the kth member $(k = 1, 2, \dots, M_0 + M)$ is identified by the N-dimensional vector $\sigma_k = (\sigma_k^1, \sigma_k^2, \dots, \sigma_k^N)$, where $\sigma_k^j = \pm 1$ is a binary variable representing the opinion of the kth member on the jth decision. For any given decision variable d_j, individuals k and h agree if $\sigma_k^j = \sigma_h^j$, otherwise they disagree.

Within the framework of Ising's approach, (Sornette, 2014), (Ising, 1925), (Brush, 1967), disagreement inside the *hierarchical* network is characterized by a certain level of conflict $\left(E_{kh}^j\right)_H$ (energy level) between the two socially interacting members k and h, i.e. $\left(E_{kh}^j\right)_H = -JI_k\sigma_k^j\sigma_h^j$, where J is the strength of the social interaction and I_k is the impact of the hierarchical status of the individual k. The hierarchical status of the individual k is associated with his/her hierarchical level $l_k \geq 1$. We assume that a disagreement with an individual belonging to a higher hierarchical level is weighted more than the disagreement with one member belonging to a lower level. This means that the probability to change opinion is higher if the individual has a different opinion with his/her chief rather than with an individual sharing the same hierarchical status. The quantity I_k is then defined as follows:

$$I_k = \exp\left(\frac{1 - l_k}{\mu}\right) \tag{5.3}$$

where the hierarchical level l_k is an integer, which increases as one moves from the top (CEO and PM) to the bottom of the hierarchy. The quantity μ instead tunes the decay rate of I_k and can be used to model the PM authority on the team members.

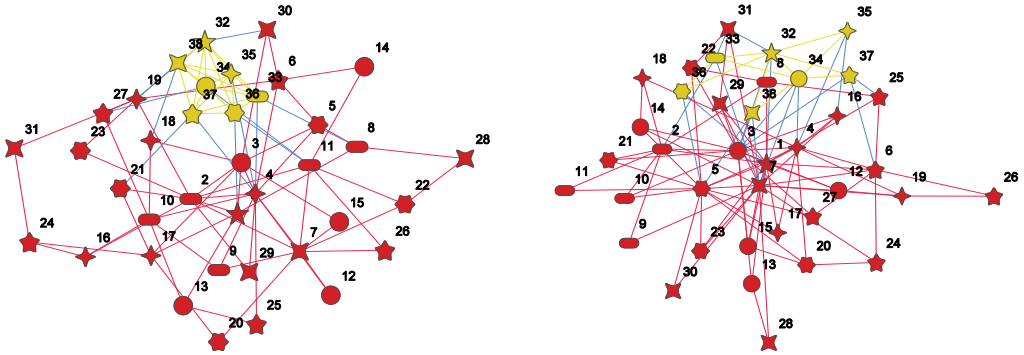
Similarly, the level of conflict on the decision j due to the relationship in the *social* network is characterized by a certain level of conflict $\left(E_{kh}^j\right)_S$ (energy level) between the two socially interacting members k and h, i.e. $\left(E_{kh}^j\right)_S = -J\sigma_k^j\sigma_h^j$.

Each agent makes his/her choices driven by the rational behavior, which pushes him to increase the perceived payoff and the social interactions, which push him to seek consensus with the others agents.

A multiplex network, (De Domenico et al., 2013), (Lee et al., 2015), (Boccaletti et al., 2014), (Wang et al., 2015), (Kivela et al, 2014), with N different layers is defined. On each layer, individuals interact on the *global* network, sharing their opinions on a certain decision variable d_j and leading to a certain level of conflict $E^j = E_H^j + E_S^j$. The graph of the *global*

network on each layer d_j is the union of the hierarchical and social graphs, i.e. its adjacent matrix is the union of the hierarchical and social network adjacent matrixes, $\boldsymbol{A}_G^j = \boldsymbol{A}_H^j \cup \boldsymbol{A}_S^j$.

In figure 5.3 examples of the *global* networks for small-world [Fig. 5.3 (a)] and scale-free [Fig. 5.3 (b)] social network are shown. The interconnections between different layers (see Fig. 4.1 of chapter 4 for analogy) represent the interactions among the opinions of the same individual k on the decision variables, while the dashed lines connecting the different decision layers represent the interaction between the opinions that each member has on the decision variables. This interaction occurs via the *NK* perceived fitness, i.e. changing the opinion on the decision variable *j* causes a modification of the perceived pay-off, which also depends on the opinions the member has on the remaining decision variables.



(a) (b)

Fig. 5.3, Examples of global networks, unions of the hierarchical and social ones, with (a) small-world and (b) scale-free connectivity of the social network. Team members are in yellow color, organization members are in red color. Blue links identify those members interacting through the external network.

Using the adjacent matrixes, we can rephrase the total level of conflict (energy level) on decision d_j as

$$E^j = E_H^j + E_S^j = -\frac{1}{2}J\{\boldsymbol{I}^j \cdot \boldsymbol{A}_H^j \boldsymbol{s}^j\} \cdot \boldsymbol{s}^j - \frac{1}{2}J\{\boldsymbol{A}_S^j \boldsymbol{s}^j\} \cdot \boldsymbol{s}^j = -\frac{1}{2}J\{\boldsymbol{B}^j \boldsymbol{s}^j\} \cdot \boldsymbol{s}^j \qquad (5.4)$$

where $\boldsymbol{B}^j = \boldsymbol{I}^j \cdot \boldsymbol{A}_H^j + \boldsymbol{A}_S^j$ and $\boldsymbol{I}^j = \{I_k\}$ for $k = 1, 2, \ldots, M_0 + M$. Observe that in this case $\boldsymbol{B}^j$ is a no symmetric real numbers matrix.

To model the dynamics of decision-making in terms of a continuous-time Markov process, we define the state vector $\boldsymbol{s}$ of size $n = (M_O + M) \times N$ as:

$$\boldsymbol{s} = (s_1, s_2, \ldots, s_n) = \left(\sigma_1^1, \sigma_2^1, \ldots, \sigma_{M_O+M}^1, \sigma_1^2, \sigma_2^2, \ldots, \sigma_{M_O+M}^2, \ldots, \sigma_1^N, \sigma_2^N, \ldots, \sigma_{M_O+M}^N\right)$$

and the block diagonal matrix $\boldsymbol{B} = \text{diag}(\boldsymbol{B}^1, \boldsymbol{B}^2, \ldots, \boldsymbol{B}^N)$. For any given *l*th component $s_l = \sigma_k^j$ of the vector $\boldsymbol{s}$ it is possible to uniquely identify the member *k* and the decision variable *j* by means of the relations $k = \text{quotient}(l - 1, M_O + M) + 1$, and $j = \text{mod}(l - 1, M_O + M) + 1$. The total level of conflict can be then rephrased as:

$$E(\boldsymbol{s}) = -\frac{1}{2}J\boldsymbol{B}\boldsymbol{s}\cdot\boldsymbol{s} = -\frac{1}{2}J\sum_{ij}B_{ij}s_i s_j \tag{5.5}$$

Observe that $B_{ii} = 0$ (with $i = 1, \dots, n$).

Now let be $P(\boldsymbol{s}, t)$ the probability that, at time *t*, the state vector takes the value $\boldsymbol{s}$ out of 2^n possible states. The time evolution of the probability $P(\boldsymbol{s}, t)$ obeys the master equation

$$\frac{dP(\boldsymbol{s}, t)}{dt} = -\sum_{l} w(\boldsymbol{s}_l \to \boldsymbol{s}_l')P(\boldsymbol{s}_l, t) + \sum_{l} w(\boldsymbol{s}_l' \to \boldsymbol{s}_l)P(\boldsymbol{s}_l', t) \tag{5.6}$$

where $\boldsymbol{s}_l = (s_1, s_2, \dots, s_l, \dots, s_n)$, $\boldsymbol{s}_l' = (s_1, s_2, \dots, -s_l, \dots, s_n)$. The transition rate $w(\boldsymbol{s}_l \to \boldsymbol{s}_l')$ is the probability per unit time that the opinion s_l flips to $-s_l$ while the others remain temporarily fixed. Recalling that flipping of opinions is governed by social interactions and self-interest a possible ansatz for the transition rates is:

$$w(\boldsymbol{s}_l \to \boldsymbol{s}_l') = \frac{1}{2}\left[1 - s_l \tanh\left(\beta J \sum_{h} B_{lh}s_h\right)\right] \times \exp\{\beta'[\Delta V(\boldsymbol{s}_l', \boldsymbol{s}_l)]\} \tag{5.7}$$

In equation (5.6) the pay-off function $\Delta V(\boldsymbol{s}_l', \boldsymbol{s}_l) = \bar{V}(\boldsymbol{s}_l') - \bar{V}(\boldsymbol{s}_l)$, where $\bar{V}(\boldsymbol{s}_l) = V_k(\sigma_k)$, is simply the change of the fitness perceived by the agent $k = \text{quotient}(l - 1, M_O + M) + 1$, when its opinion $s_l = \sigma_k^j$ on the decision $j = \text{mod}(l - 1, M_O + M) + 1$ changes from $s_l = \sigma_k^j$ to $s_l' = -\sigma_k^j$.

The quantity β is instead the inverse of the so-called social temperature and is a measure of the chaotic circumstances, which lead to a random opinion change. The term β' is related to the degree of confidence the agents have about the perceived fitness (the higher β' the higher the confidence). To solve the Markov process equations (5.6) and (5.7), we employ a simplified version of the exact stochastic simulation algorithm proposed by Gillespie (Gillespie, 1976-1977), see also Appendix 8.4.

The group fitness value equation (5.1) and the level of agreement between the team members (i.e. social consensus) are used to measure the performance of the collective-decision making process. To calculate the group fitness value, the vector $\boldsymbol{d} = (d_1, d_2, \dots, d_N)$ needs to be determined. To this end, consider the set of opinions $(\sigma_1^j, \sigma_2^j, \dots, \sigma_M^j)$ that the team members have about the decision *j*, at time *t*. The decision d_j is obtained by employing the majority rule, i.e. we set

$$d_j = \text{sgn}\left(M^{-1}\sum_{k=1}^{M}\sigma_k^j\right), \quad j = 1, 2, \dots, N. \tag{5.8}$$

If M is even and in the case of a parity condition, d_j is, instead, uniformly chosen at random between the two possible values ± 1. The group fitness is then calculated as $V[\boldsymbol{d}(t)]$ and the ensemble average $\langle V(t)\rangle$ is then evaluated. The efficacy of the group in optimizing $\langle V(t)\rangle$ is then calculated in terms of normalized average fitness

$$\eta(t) = \frac{\langle V(t)\rangle - V_{min}}{V_{max} - V_{min}} \tag{5.9}$$

where $V_{min} = \min[V(\boldsymbol{d})]$ and $V_{max} = \max[V(\boldsymbol{d})]$.

Similarly to the DMM, the entire consensus of the group on the whole set of decisions is:

$$\chi(t) = \frac{1}{N}\sum_{j} \chi^j(t) = \frac{1}{M^2 N}\sum_{j=1}^{N}\sum_{hk=1}^{M} R_{hk}^{j}(t) \tag{5.10}$$

Note that $0 \leq \chi(t) \leq 1$.

5.2 Simulation and results

We consider, unless differently specified, a team composed of $M = 7$ agents with an average knowledge level, $p = 0.5$, and an organization composed of $M_O = 31$ agents, subdivided into 6 departments, headed by their functional manager. Each department has 2 hierarchical levels (functional managers and operating units) and 4 links per node.

Two social networks are defined; the first one, *external*, involves all the agents (team and organization), the second one, *internal*, involves only the team members. In the simulations, we assumed that both external and internal social networks have the same topology, i.e. both small-world or both scale-free.

The external social network has a density $d_{ext} = 0.1$ and in the case of a small-world network, the value of the rewiring probability used is $r_p = 0.001$. The internal social network has a density $d = 0.\overline{3}$ and in the small-world case a rewiring probability $r_p = 0.001$.

A $N = 12$, $K = 11$ landscape has been used so that the agents of each one of the 6 departments have expertise and knowledge about 2 decisions. We also set $\beta' = 10$, since we assume that the agents have a good confidence about their perceived fitness function. Vespignani (Leone, Vázquez, Vespignani, & Zecchina, 2002) and Mendes (Dorogovtsev, Goltsev, & Mendes, 2002) demonstrated that for general graphs, the critical transition of the Ising model occurs at:

$$(\beta J)_c = -\frac{1}{2}\log\left(1 - 2\frac{\langle k\rangle}{\langle k^2\rangle}\right) \tag{5.11}$$

where $\langle k\rangle$ and $\langle k^2\rangle$ are respectively the mean degree and the mean squared degree of the team global network of interaction. As we have seen in chapter 4, the threshold value $(\beta J)_c$ for the DMM, is affected by the values of β', nevertheless, it can be fine approximated by equation (5.11) for the case of study, in which $\beta' = 10$. We define a parameter $\rho = \beta J/(\beta J)_c$, i.e. $\rho < 1$ $(\rho > 1)$ describes a subcritical (supercritical) level of interactions among agents.

We simulate many scenarios to investigate the influence of the parameter ρ on the final outcome of the decision-making process. The simulation is stopped at steady-state. This condition is identified by simply taking the time-average of team consensus and pay-off over consecutive time intervals of fixed length T and by checking that the difference between two consecutive averages is sufficiently small. For any given scenario, each stochastic process equations (5.6) and (5.7) is simulated by generating *100* different realizations (trajectories).

Figure 5.4 shows the steady-state values of the normalized averaged team fitness $\eta_\infty = \frac{\langle V(\mathrm{t}\to\infty)\rangle - V_{min}}{V_{max}-V_{min}}$ [Fig. 5.4 (a, d)], social consensus $\chi_\infty = \chi(\mathrm{t}\to\infty)$ [Fig.5.4 (b, e)] and simulation time τ [Fig.5.4 (c, f)] as a function of the control parameter $\rho = \beta J/(\beta J)_c$, for the two topologies of the social networks considered, small-world and scale-free, and for a *hierarchical* and *flat* team (presence or not of the PM). In the hierarchical case, three different knowledge values of the PM have been investigated ($p_{PM} = 0.1, 0.5, 1.0$).

For both the kind of connectivity, we observe that the best performance of the team in terms of averaged final payoff are obtained around the critical value of interaction $(\rho = 1)$. In the subcritical regime, $(\rho < 1)$, the decision-making process is inefficient because each team member makes his/her choices to optimize the perceived fitness, but poorly interacting with the other members, he/she behaves almost independently, not receiving a sufficient feedback on the wisdom of his/her choices. At the criticality, consensus emerges, so agents share knowledge and information, finding an agreement on their choices. This brings to an evident improvement of the final payoff. Moving to the supercritical regime $(\rho > 1)$ the strength of social interactions becomes too high and the consensus among members is rapidly achieved, not allowing a proper exploration of the fitness landscape. This carries the performance of the decision-making process worsening. It is worth nothing that the utilization of equation (5.11) for appropriately calculate the critical value of the interaction strength is confirmed

by the simulation time τ [Fig.5.4 (c, f)] , always maximal at the critical value $(\rho = 1)$, as predicted in literature.

Let us concentrate now on the effect of the social network topology on the performances reached. Small-world networks with low rewiring probability are similar to cyclical networks, where each agent is only connected to its neighbors. In these circumstances the PM in the hierarchical network, allows an indirect interaction between agents far from each other, that leads to an improvement of the performance in terms of final payoff. Nevertheless, compared to the case of a flat team, a hierarchical one, requires a longer time to find a final shared solution. The higher the PM knowledge, the higher is the performance reached around the critical threshold, nevertheless, the difference in terms of normalized group fitness η_∞ becomes already negligible moving from $p_{PM} = 0.5$ to $p_{PM} = 1.0$.

In Scale-free networks only a few agents are very connected, who become hubs of the network, through which all agents can indirectly and rapidly communicate. In this kind of network the PM in the hierarchical network does not particularly influence the performance. Also in this case, the previously observed slight effect of his/her knowledge on the payoff outcome is confirmed.

Finally, we can state that the main difference between the two social networks topologies, in terms of final performance, consists in the effect of the PM presence in the hierarchical network, fundamental for the small-world case to get higher performance, or not for the scale-free one. A notable difference is also observed about the time requires reaching the convergence: the brainstorming process of scale-free network is generally faster, compared with the one of a small-world network.

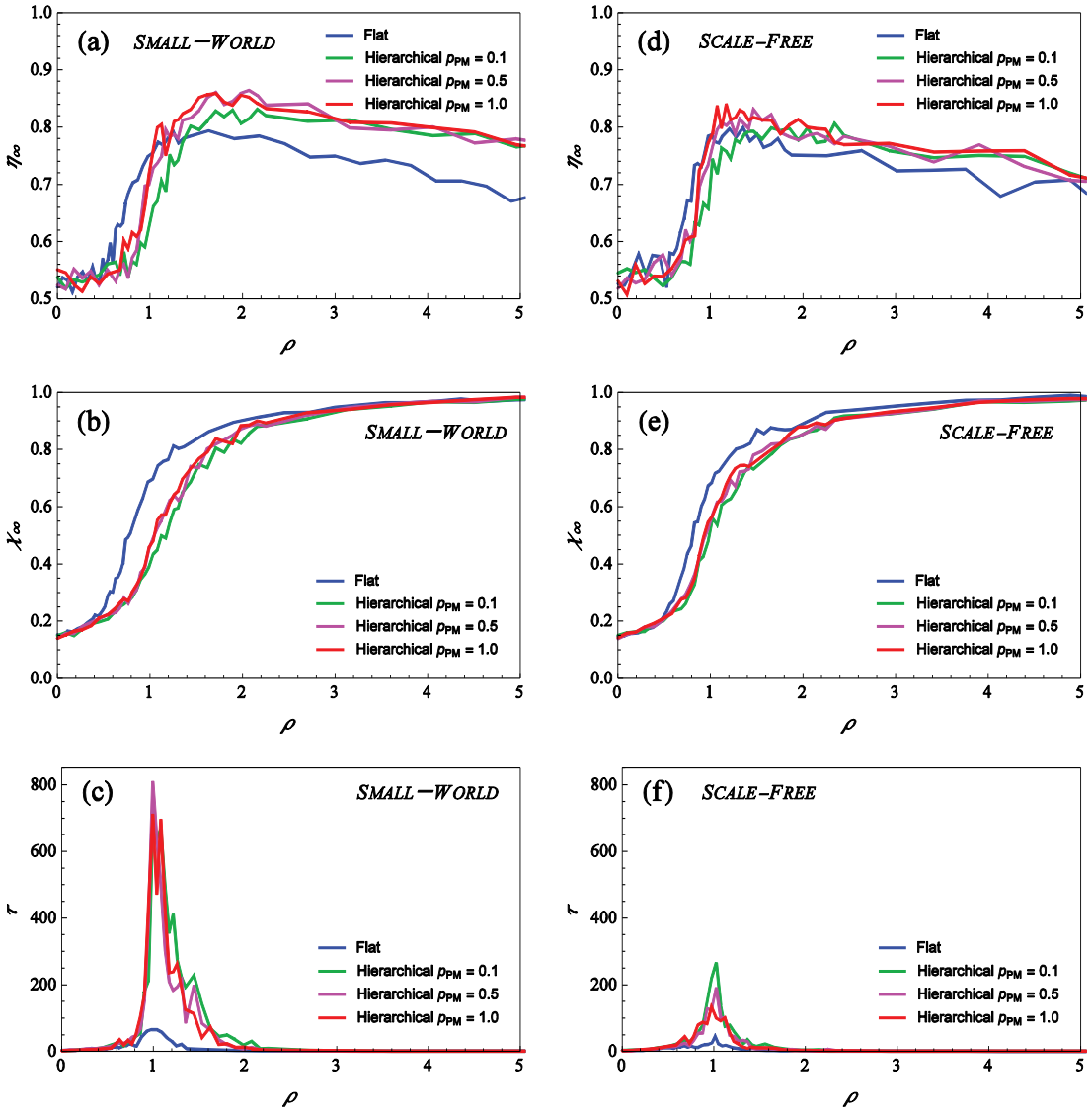


Fig. 5.4, The stationary values of the (a, d) normalized averaged fitness η_∞, (b, e) social consensus χ_∞ and (c, f) simulation time τ, as a function of the control parameter $\rho = \beta J/(\beta J)_c$, for small-world and scale-free connectivity of the social network. Results for a hierarchical structure of the team, considering also a different knowledge of the PM, and a flat structure, are shown.

We also investigated the effect of the density d of the team social network on its performance. Increasing d, the mean degree $\langle k \rangle$ increases and $(\beta J)_c$ decreases, according to the Vespignani-Mendes formula, eq. (5.10). Working at a fixed value of strength interaction βJ, i.e. $\beta J = 0.20$ in fig. 5.5, the system passes from a *subcritical* to a *supercritical* situation, with a transition clearly visible on the performance. The critical density value is identified by the vertical dashed line in the diagrams. It is worth nothing that the limited amount of density values explored is due to the small number of agents considered ($M = 7$) and to the social network generation strategy.

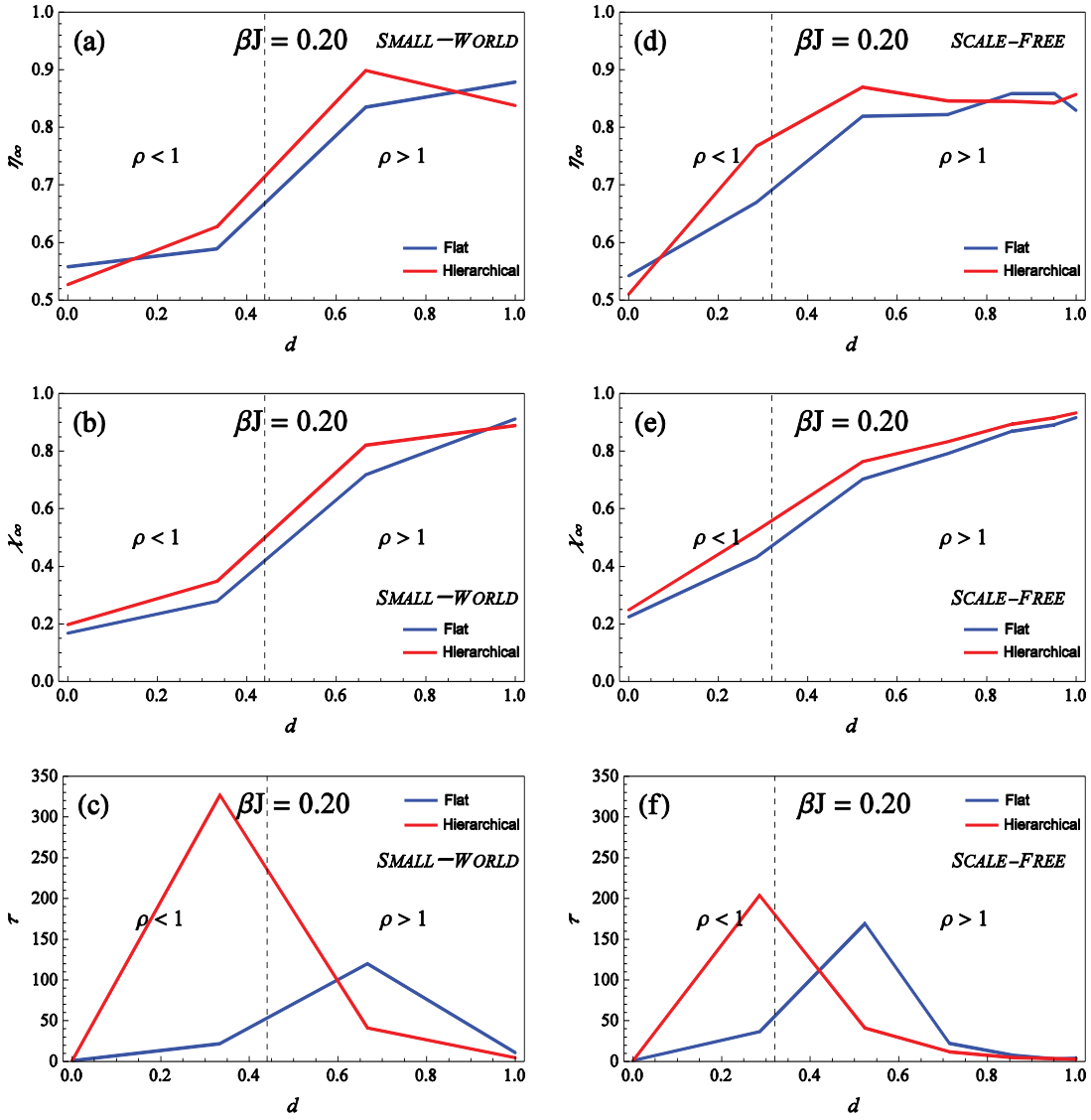


Fig. 5.5, The stationary values of the (a, d) normalized averaged fitness η_∞, (b, e) social consensus χ_∞ and (c, f) simulation time τ, as a function of the density d of the social network, with small-world or scale-free connectivity. Results for a hierarchical and flat structure of the team are shown.

Similarly, increasing the team size but working at a fixed team social network density, i.e. $d = 0.\overline{3}$, the mean degree $\langle k \rangle$ increases and $(\beta J)_c$ decreases, according to the Vespignani-Mendes formula, eq. (5.10). Working at a fixed value of βJ, i.e. $\beta J = 0.12$ in figures 5.6, the same kind of transitional behavior is observed. This result is quite interesting because it is telling us that, given a level of interactions between agents and a density of the social network, an optimal size of the team exist, in order to maximize its performance.

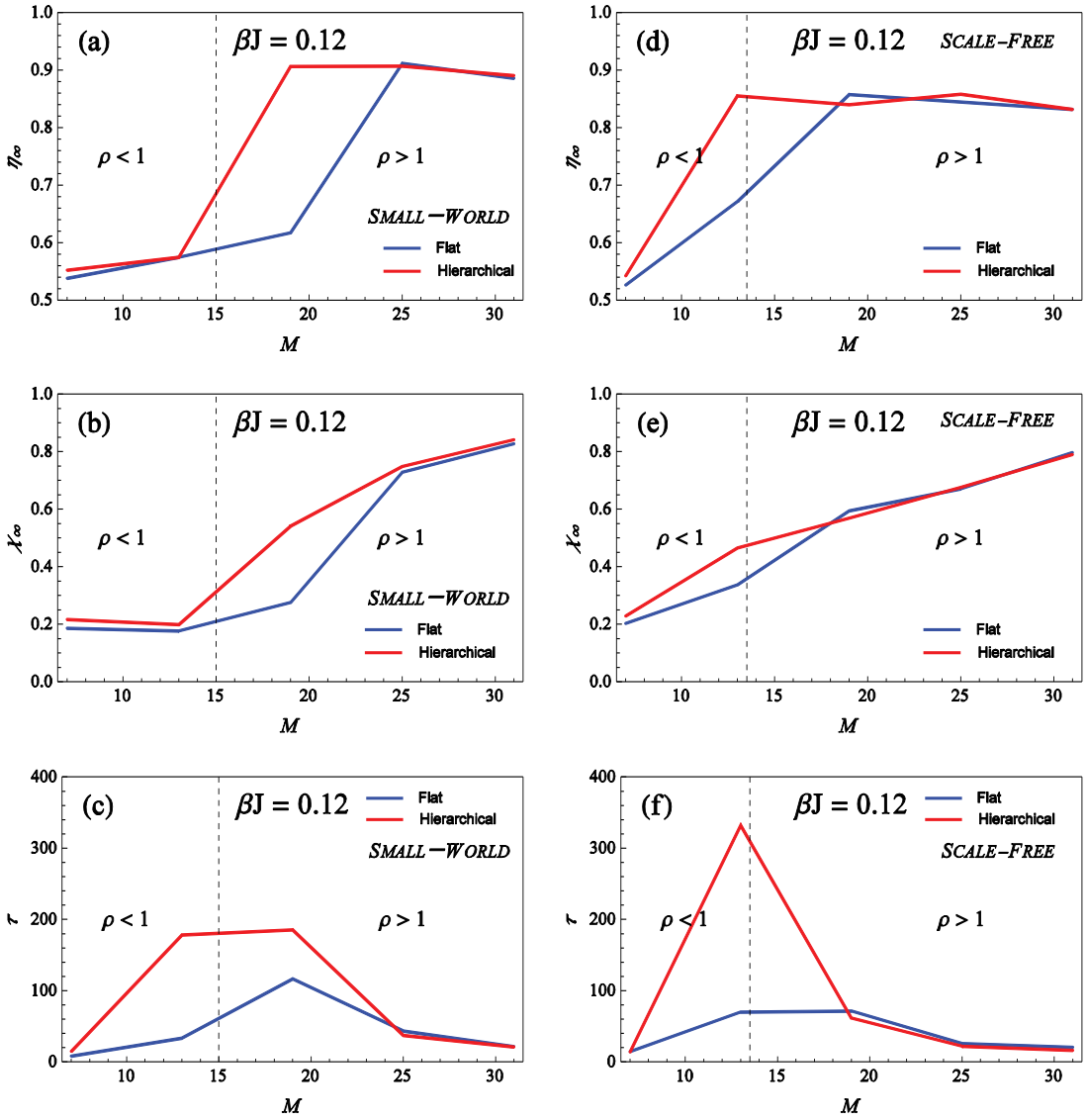


Fig. 5.6, The stationary values of the (a, d) normalized averaged fitness η_∞, (b, e) social consensus χ_∞ and (c, f) simulation time τ, as a function of the number of team agents M, for small-world and scale-free connectivity of the social network. Results for a hierarchical and flat structure of the team are shown.

In conclusion, we analyzed the effect of the PM authority on the performance of the team in the case in which his knowledge level is poor, $p_{PM} = 0.1$. We model the team authority using the parameter μ in equation (5.3). We use $\mu = 1$ for describing a PM with a strong authority on the team members and $\mu = \infty$ for one with no authority. The internal team social network has a density $d = 0.\overline{3}$ and in the small-world case a rewiring probability $r_p = 0.001$. Results in Fig. 5.7, illustrate that a team guided by a PM with low knowledge (p_M=0.1) and strong authority ($\mu = 1$) on the others team members shows a smoother critical

transition and essentially similar performance in terms of payoff and consensus, but in a longer time, compared with the case in which the PM has not authority ($\mu = \infty$).

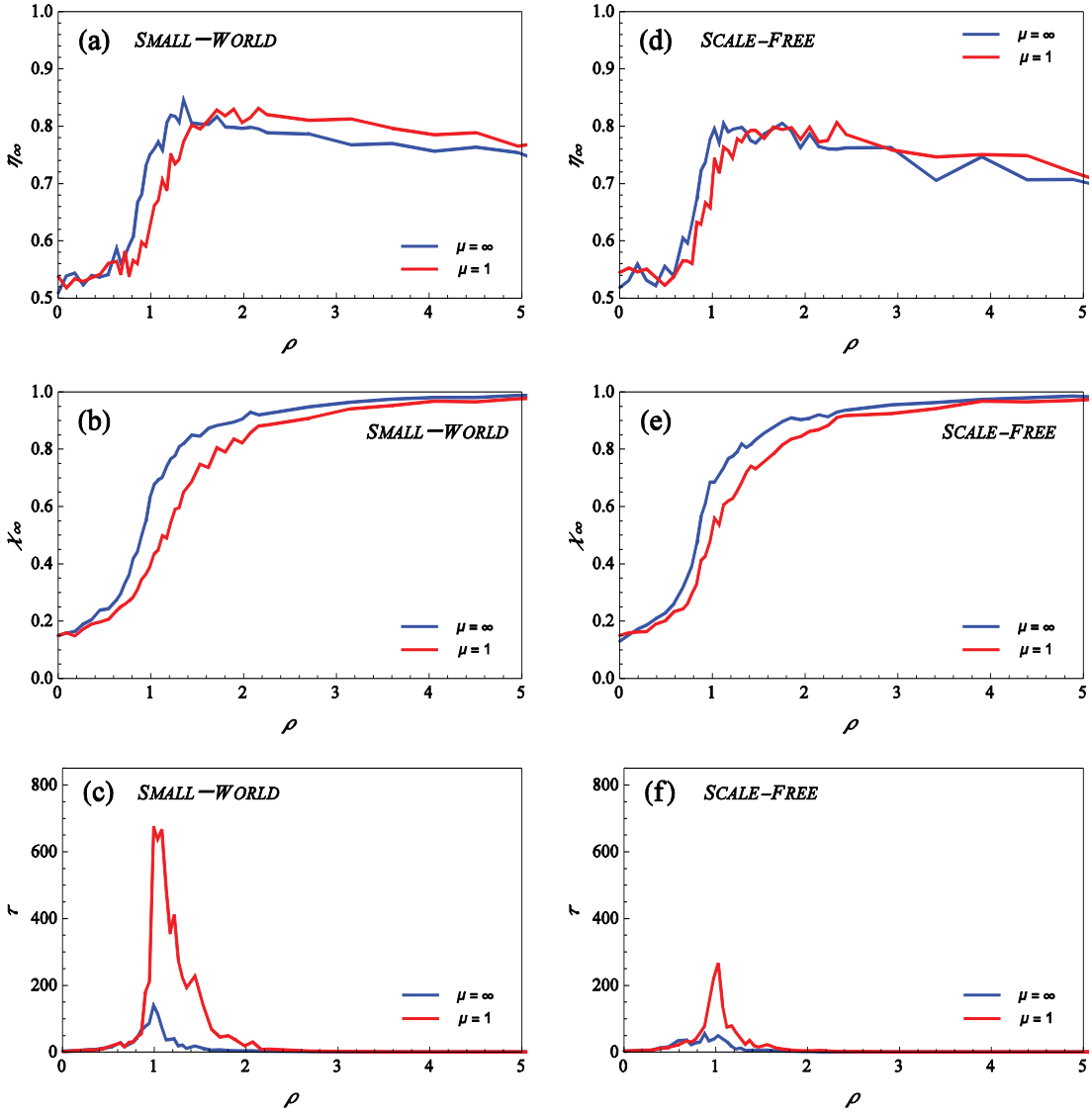


Fig. 5.7, The stationary values of the (a, d) normalized averaged fitness η_∞, (b, e) social consensus χ_∞ and (c, f) simulation time τ as a function of the control parameter $\rho = \beta J/(\beta J)_c$, for small-world and scale-free connectivity of the social network. The team structure is hierarchical, with a PM characterized by a strong ($\mu = 1$) or not ($\mu = \infty$) authority on the team members.

5.3 Conclusions

In this chapter, the author presents an application of the decision-making model (DMM) proposed in chapter 4, to the simulation of a management problem, in which a team of individuals is in charge to design a new product. We suppose the interactions among individuals take place for hierarchical or social reasons, investigating respectively the influence of the team hierarchical structure and of social network topology, on the team performance. Two kinds of social network connectivity have been considered, small-world and scale-free. We simulate how team members converge towards a shared solution of the design process by interacting with each other through hierarchical and social networks.

We found that a critical value of interaction, easily predictable using the Vespignani-Mendes formula allows for knowledge transfer among the team members and better performance of the decision-making process. The main difference between the two social networks topologies, in terms of final performance, consists in the effect of the PM presence in the hierarchical network, fundamental for the small-world case to get higher performance, and not for the scale-free one. A remarkable difference is also observed about the time requires reaching the convergence, smaller for scale-free network compared to the small-world ones.

We also studied the effects of the density of the social network and of the team size on its performance, noticing that working with fixed values of social interaction and team size (social network density), an optimal value of the social network density (team size) exists.

In conclusion, we investigate the effect of the project manager authority on the team performance, finding that the only considerable difference is in the time required to reach a solution.

It is worth noting that the application proposed is just an example, considerable interesting in the management field for the results obtained, nevertheless is evident how following a similar formulation a huge number of decision-making problems, involving for example parliaments, squads of soldiers, coordination of human crowds, etc. can be easily modeled.

6 Human Group Optimization algorithm

"All models are wrong,
but some are useful."
George E. P. Box

A large number of optimization algorithms have been developed by researchers to solve a variety of complex problems in operations management area. In this chapter, the author presents a novel optimization algorithm belonging to the class of swarm intelligence optimization methods. The algorithm mimics the decision-making process of human groups and exploits the dynamics of this process as an optimization tool for combinatorial problems. To achieve this aim, a continuous-time Markov process is proposed to describe the behavior of a population of socially interacting agents, modeling how humans in a group modify their opinions driven by self-interest and consensus seeking. As in the case of a collection of spins, the dynamics of such a system is characterized by a phase transition from low to high values of the consensus (magnetization). We recognize this phase transition as being associated with the emergence of a collective superior intelligence of the population. While this state being active, a cooling schedule is applied to make agents closer and closer to the optimal solution, while performing their random walk on the fitness landscape. A comparison with simulated annealing as well as with a multi-agent version of the simulated annealing and genetic algorithms is presented in terms of efficacy in finding a good solution on an *NK*-Kauffman landscape. In all cases, our method outperforms the others, particularly in presence of limited knowledge of the agents.

The chapter is organized as follows. In Sec. 6.1 a brief classification of the optimization algorithms developed so far and their core applications is provided. In Sec. 6.2 the author presents the Human Group Optimization (HGO) algorithm and its main features. Sec. 6.3 tests the HGO algorithm in solving *NP*-complete problems of increasing complexity and compares it with the Simulated Annealing, a Multi-Agent version of Simulated Annealing and Genetic Algorithms. In Sec. 6.6 the main conclusions are drawn, discussing also future perspectives.

6.1 Introduction

Researchers have developed many meta-heuristic algorithms inspired by nature with the aim of solving combinatorial optimization problems. A common way to classify them is to distinguish between *trajectory* and *population-based* algorithms.

Trajectory algorithms, such as Simulated Annealing (SA) (Kirkpatrick, Gelatt, Vecchi, & others, 1983) and Quantum Annealing (Kadowaki & Nishimori, 1998), (Farhi, et al., 2001), describe a trajectory (usually a random walk) in the search space to reach the solution.

Population-based algorithms perform multiple search processes, each of them carried out by a different agent. Population-based algorithms can be further distinguished in two classes: *Evolutionary algorithms* and *Swarm-based algorithms* (Binitha & Sathya, 2012).
The evolutionary algorithms, of which the genetic algorithm is most popular (Simon, 2013), mimic the processes of natural evolution, such as mutation, selection, and inheritance, to identify the best solution. The swarm algorithms exploit the collective intelligence of the social groups, such as flock of birds, ant colonies, and schools of fish, in accomplishing different tasks. They include the Ant Colony Optimization (Dorigo M. , 1992), (Dorigo, Maniezzo, & Colorni, 1996), (Dorigo & Gambardella, 1997), the Particle Swarm Optimization (Kennedy, 2011), the Differential Evolution (Storn & Price, 1997), the Artificial Bee Colony (Karaboga, 2005), (Karaboga & Basturk, 2007), the Glowworm Swarm Optimization (Krishnanand & Ghose, 2009), the Cuckoo Search Algorithm (Yang & Deb, 2010), and very recently the Grey Wolf Optimizer (Mirjalili, Mirjalili, & Lewis, Grey wolf optimizer, 2014) and the Ant Lion Optimizer (Mirjalili, 2015).

All these algorithms have been thoroughly employed to deal with *NP*-complete optimization problems, e.g. Knapsack problem, Makespan Scheduling, Travelling Salesman Problem (TSP), Clique problem and others.
For example, Simulated Annealing has been mainly employed to solve the traveling salesman problem (Golden & Skiscim, 1986), scheduling problems (Radhakrishnan & Ventura, 2000), (Melouk, Damodaran, & Chang, 2004), facility location and supply chain design problems (Arostegui, Kadipasaoglu, & Khumawala, 2006).
The Genetic Algorithms count a larger number of applications compared to Simulated Annealing, even though the wideness of the areas to which they have been applied is quite narrow (Chaudhry & Luo, 2005). Aytug et al. (Aytug, Khouja, & Vergara, 2003) provide an interesting review of the use of genetic algorithms for solving different types of

operations problems including production control, facility layout design, line balancing, production planning, and supply chain management.

The last years have seen a huge growth of the applications of swarm-based algorithms in the field of Swarm Intelligence, which is the specific branch of Artificial Intelligence, dealing with the collective behavior of swarms through the complex interaction of individuals with no centralized coordination. Swarm optimization algorithms share remarkable features, such as decentralization, self-organization, autonomy, flexibility, and robustness, which have been proven very useful to solve complex operational tasks (Ottino, 2004), (Bonabeau, Dorigo, & Theraulaz, 1999).

Applications of ant colony optimization algorithm mainly concern the traveling salesman problem, scheduling, vehicle routing, and sequential ordering (Dorigo, Di Caro, & Gambardella, 1999). More recently, they have been also employed in supply chain contexts to solve production-inventory problems (Ferretti, Zanoni, & Zavanella, 2006), (Nia, Far, & Niaki, 2014) and network design (Moncayo-Martinez & Zhang, 2011).

These algorithms reproduce the collective decision process that makes social groups superior in solving tasks compared to single individuals. Agents (ants, bees, termites, fishes) make choices, pursuing their individual goals (forage, survive, etc.) based on their own knowledge and amount of information (position, sight, etc.), and adapting their behavior to the actions of the other agents. The group-living enables social interactions to take place as a mechanism for knowledge and information sharing (Couzin, 2009), (Sumpter & Pratt, 2009), (Ward, Sumpter, Couzin, Hart, & Krause, 2008), (Arganda, Pérez-Escudero, & de Polavieja and Gonzalo, 2012), (Ward, Herbert-Read, Sumpter, & Krause, 2011), (Pérez-Escudero & Polavieja, 2011), (Watts, 2002), (Turalska, Lukovic, West, & Grigolini, 2009), (Wang, Szolnoki, & Perc, Interdependent network reciprocity in evolutionary games, 2013), (Wang, Szolnoki, & Perc, 2013). Even though the single agents may possess a limited knowledge, and their actions are usually very simple, the collective behavior, enabled by the social interactions, leads to the emergence of a superior intelligence of the group.

In this chapter, the author proposes a novel swarm intelligence optimization algorithm to solve complex combinatorial problems. The proposed algorithm is inspired by the behavior of human groups and their ability to solve a very large variety of complex problems, even when the individuals may be characterized by cognitive limitations. Although it is widely recognized that human groups, such as organizational teams, outperform single individuals

in solving many different tasks including new product development, R&D activities, production and marketing issues, the authors are not aware of the presence in the scientific literature of optimization algorithms inspired by the problem-solving process of human groups. Similarly to other social groups, human groups are collectively able, by exploiting the potential of social interactions, to achieve much better performance than single individuals can do. This specific ability of human groups has been defined as *group collective intelligence* (Woolley, Chabris, Pentland, Hashmi, & Malone, 2010), (Engel, Woolley, Jing, Chabris, & Malone, 2014) that recently is receiving a growing attention in the literature as to its antecedents and proper measures (Woolley, Chabris, Pentland, Hashmi, & Malone, 2010), (Engel, Woolley, Jing, Chabris, & Malone, 2014).

The proposed algorithm, hereafter referred to as Human Group Optimization (HGO), is developed within the methodological framework recently proposed by Carbone and Giannoccaro (Carbone & Giannoccaro, 2015) to model the collective decision making of human groups. This model captures the main drivers of the individual behavior in groups, i.e., self-interest and consensus seeking, which should lead to the emergence of collective intelligence. The group is conceived as a set of individuals making choices based on rational calculation and self-interested motivations. However, any decision made by the individual is also influenced by the social relationships he/she has with the other group members. This social influence pushes the individual to modify the choice he/she made, for the natural tendency of humans to seek consensus and avoid conflict with people they interact with (Dimaggio & Powell, 1983). As a consequence, effective group decisions spontaneously emerge as the result of the choices of multiple interacting individuals. Herewith, we identify which circumstances lead to the emergence of the collective intelligence of the group. We show that by tuning some control parameters it is possible to make the system undergo a critical transition towards a state of high consensus which is always accompanied by an analogues transition from low to high group fitness values. We find that at the critical transition the flow of information from the fitness landscape to the group of agents is maximized, thus improving the abilities of the group as a whole to explore the fitness landscape searching for the optimal solution. To test the ability of the HGO algorithm, we compare its performance with those of the Simulated Annealing (SA), and Genetic Algorithm (GA) in solving NP-complete problems, consisting in finding the optimum on a fitness landscape, the latter generated within the Kauffman *NK* model of complexity (Kauffman & Levin, 1987), (Kauffman & Weinberger, 1989).

6.2 The Human Group Optimization algorithm

In this section, the author designs the HGO algorithm, by exploiting the collective intelligence property of the decision-making model (DMM) (Carbone & Giannoccaro, 2015) developed so far to solve discrete NP combinatorial problems (chapter 4).

To this aim, the process followed to design the Simulated Annealing algorithm (Kirkpatrick, Gelatt, Vecchi, & others, 1983) is emulated. We first observe that the Markov process defined in Eq. (4.5) with transitions rates Eq. (4.6) converges to the stationary probability distribution (Appendix 8.3)

$$P_0(\boldsymbol{s}_l) = \frac{\exp[-\beta E(\boldsymbol{s}_l) + 2\beta' \bar{V}(\boldsymbol{s}_l)]}{\sum_k \exp[-\beta E(\boldsymbol{s}_k) + 2\beta' \bar{V}(\boldsymbol{s}_k)]} \tag{6.1}$$

where the total level of conflict is $E(\boldsymbol{s}) = -0.5\langle k \rangle^{-1} J \sum_{ij} A_{ij} s_i s_j$. Eq. 6.1 is a Boltzmann distribution with effective energy

$$E_{\text{eff}}(\boldsymbol{s}_l) = -\bar{V}(\boldsymbol{s}_l) + \alpha E(\boldsymbol{s}_l) \tag{6.2}$$

where $\alpha = \beta/(2\beta')$. We then make the parameters βJ and β' change during the process as follows:

$$\begin{cases} \beta' = \beta_0' \log(i+1) \\ \beta J = \min\{\mu(i-1), (\beta J)_c\} \end{cases} \tag{6.3}$$

where i is the time iterator, μ is a free parameter tuning the rate of the interaction strength growth law, and β_0' is set per Ref. (Ben-Ameur, 2004). The threshold value $(\beta J)_c$ is properly chosen to guarantee that the critical transition to the collective intelligence state is always completed during the process. Also, the procedure leads the ratio $\alpha = \beta/(2\beta')$ to vanish in the long-term limit to allow $E_{\text{eff}}(\boldsymbol{s}_l) \to -\bar{V}(\boldsymbol{s}_l)$. Note that, when agents possess complete knowledge $(p = 1)$, the latter condition, akin the Simulated Annealing, makes the proposed algorithm converge in probability to the optimum of $V(\boldsymbol{d})$ (Geman & Geman, 1984), (Hajek, 1988).

Also, observe that the choice $\beta J = 0$ identifies an optimization algorithm very closely related to the Simulated Annealing, except that the fitness landscape is explored by M non-interacting agents. Hereafter, this algorithm will be referred to as Multi-Agent Simulated Annealing (MASA). Observe that MASA is characterized by the absence of social interactions among the agents, and, as such, it is unable to exploit the collective intelligence properties of the group.

6.3 Simulation and results

In this section, we will analyze the performance of the HGO algorithm for the case of an *NK* Kauffman landscape with two different values of $N = 27$, and 100, and K ranging from 11 to 23. Such landscapes are very complex as the entire *NK* fitness landscape can be constructed (see appendix 8.1) by combining, through the Kaufmann determinist rule, $L = 2^{K+1}N$ different real values, drawn at random from a uniform distribution. We can estimate an upper bound for the combinatorial complexity C of the landscape, in a Kolmogorov sense, as

$$C = \log_2 L = K + 1 + \log_2 N \tag{6.4}$$

Eq. (6.4) shows that the parameter K is much more influential than N in affecting the complexity of the landscape. It is worth noticing that for $K = N - 1$ the complexity becomes

$C = N + \log_2 N$, so that L increases exponentially with N. In this case, we also expect that the number of local optima exponentially increases with N, as indeed found by Kaufmann (Kauffman & Levin, 1987), (Kauffman & Weinberger, 1989), who showed that for $K = N - 1$ the number of local optima evolves on the average as $2^N/(N + 1)$.

We now consider the optimization problem consisting in finding the global optimum on different *NK* fitness landscapes. In all simulations, each stochastic process is simulated by generating 500 different realizations and the ensemble average of the results is calculated. The simulation is stopped at steady-state, i.e., when changes in the time-averages of consensus and pay-off over consecutive time intervals of a given length is sufficiently small. Recall that $V(\boldsymbol{d}) = N^{-1}\sum_k W_k$ is calculated as the average of statistically independent contributions W_k uniformly drawn in the interval [0,1]. Therefore, for large N the distribution of $V(\boldsymbol{d})$ must converge in probability to a Gaussian distribution of mean $\bar{V} = \bar{W}$ (where $\bar{W} = 0.5$ is the expected value of each contribution W_k), and variance $\sigma_V^2 = \langle (V - \bar{V})^2 \rangle = N^{-1}\sigma_W^2$, where σ_W^2 is the variance of each contribution W_k. Note that this property holds for all K values as the correlation between the decisions d_j introduced by K is not important when considering individual fitness values. Thus, in order to compare the performance of the HGO algorithm on different *NK* landscapes we need to make these landscapes comparable. Hence, we define the rescaled fitness as (Appendix 8.7)

$$V \to \bar{V} + \sqrt{N}(V - \bar{V}) \tag{6.5}$$

The rescaling makes V have always the same variance σ_W^2 not depending on N, provided N sufficiently large.

Fig. 6.1 reports the HGO outcomes in terms of the time-evolution (i is the time iterator) of the fitness values $\langle V(t)\rangle$ and consensus $\chi(t)$. Calculations have been carried out for $N = 27$, $K = 11, 17, 23$. The number of agents is $M = 11$. Different levels of knowledge ranging from $0.1 \leq p \leq 1$ have been considered.

The initial parameters $\beta_0' = T_0^{-1}$, i.e. the level of confidence of the agents about their perceived fitness at the beginning of the simulation process (correspondent the inverse initial temperature of the Simulated Annealing, see Appendix 8.5), are 1.03, 0.83 and 0.72. They are computed assuming an initial acceptance probability of 0.8, respectively to the three *NK* landscape of increasing complexity.

We observe that independently of the complexity level K, the increase of fitness $\langle V(t)\rangle$ is always accompanied by a simultaneous increase of the consensus $\chi(t)$. This confirms that, as required by the developed methodology, during the optimization process, the critical transition to the collective intelligence state of the system always occurs.

We note that the complexity parameter K influences the performances of the optimization method. In fact, increasing K, i.e. increasing the complexity of the landscape, only slightly reduces the performance of the optimization algorithm (see also Fig. 6.3). In particular, for $p = 1.0$, and $N = 27$, values of $K = 11, 17, 23$ lead respectively to final fitness values $V_\infty = 1.85, 1.75, 1.63$. In terms of distance from the average fitness $\bar{V} = 0.5$, these values correspond to $V_\infty = 4.7\sigma_W, 4.3\sigma_W, 3.9\sigma_W$ respectively, which are about five times the standard deviation of the fitness landscape.

Results show that improving the knowledge of the members, i.e. increasing p, enhances the performance of the optimization process. A higher steady-state fitness V_∞, and a faster convergence toward the steady state are observed. Note also, that increasing p above 0.2 reduces the fluctuations of $\langle V(t)\rangle$, because of the higher agreement achieved among the members at higher level of knowledge.

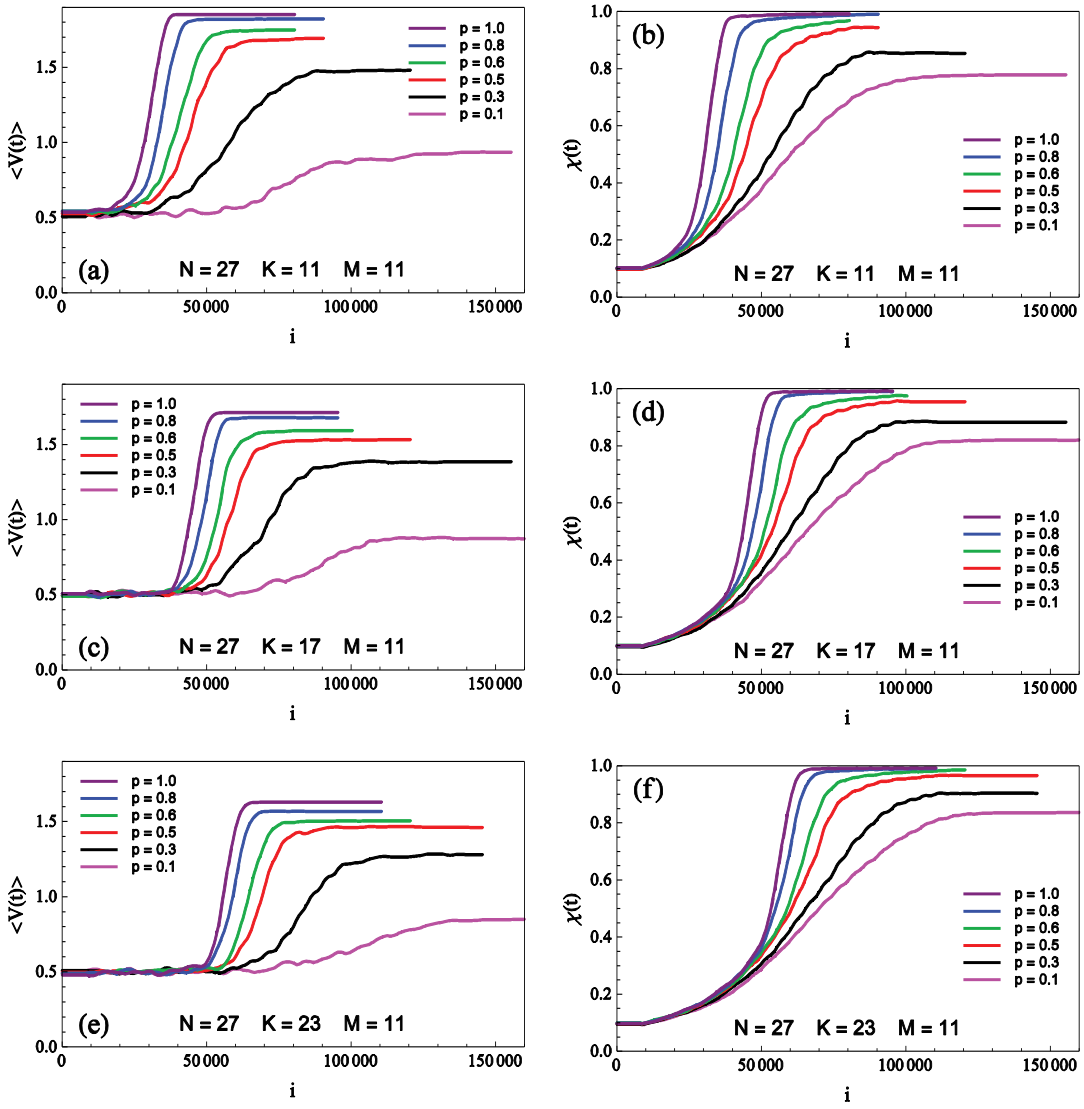


Fig. 6.1, The time-evolution of the average group fitness $\langle V(t)\rangle$ and statistically averaged consensus $\chi(t)$, for $p = 0.1, 0.3, 0.5, 0.6, 0.8, 1.0$, $M = 11$, $N = 27$, (a, b) $K = 11$, (c, d) $K = 17$, (e, f) $K = 23$.

Fig. 6.2 reports the time evolution $\langle V(t)\rangle$ and consensus χ(t) for the case $N = 100$, $K = 15$, and $M = 11$. In this case β_0', assuming the same initial acceptance probability of 0.8, is 1.67.

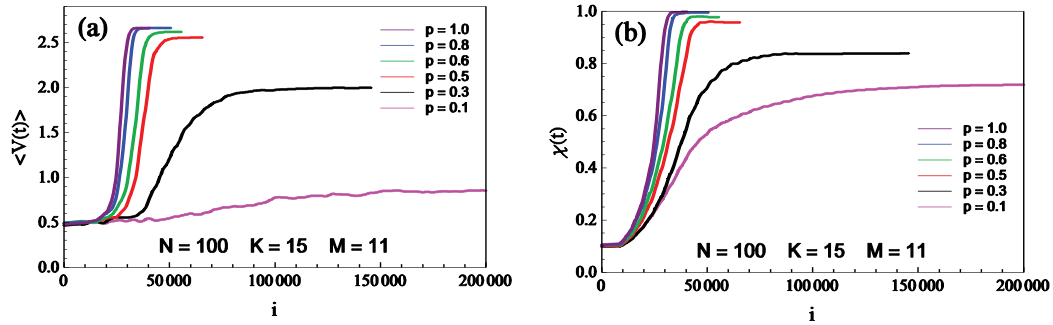


Fig. 6.2, The time-evolution of the average group fitness $\langle V(t)\rangle$ and statistically averaged consensus $\chi(t)$, for $p = 0.1, 0.3, 0.5, 0.6, 0.8, 1.0$, $M = 11$, $N = 100$, $K = 15$.

The level of knowledge p of the agents, instead, may significantly affect the performance of the optimization algorithm. Fig. 6.3, reports the steady-state values of the efficacy V_∞ [Fig. 6.3 (a)], and consensus χ_∞ [Fig. 6.3 (b)], as a function of the level of knowledge p, for $N = 27$, $K = 11, 17, 23$, and $M = 11$.

As already observed, we note that increasing the fitness landscape complexity causes an almost negligible efficacy deterioration in all the range of p.

Concerning, the effect of p on the HGO performance at given K, the diagrams in Fig. 6.3 (a), shows that the fitness values increase with p but at a different rate. Indeed, there is a region of knowledge values $p \leq 0.4$ where the V_∞ increase occurs at the highest rate. Above this interval the performance negligibly changes, being very good for $p > 0.4$, which seems to be a threshold value that must be exceeded to guarantee a high degree of consensus χ_∞ [Fig. 6.3 (b)] among the agents and, in turn, high fitness values [Fig. 6.3 (a)]. As noticed for the DMM [Fig. 4.5], this indicates that the knowledge of the agents is subjected to a saturation effect: a moderate level of knowledge is already enough to guarantee very good performance of the optimization process, while higher knowledge levels being only needed to accelerate the convergence of the optimization process.

Also, the trend of the final level of consensus [Fig. 6.3 (b)] results negligibly influenced by K. For vanishing values of p the consensus χ_∞ takes high values, as each agent's choice is driven only by consensus seeking. Increasing p initially causes a decrease of consensus, as the self-interest of each member leads to a certain level of disagreement. However, a further increment of p makes the members' knowledge overlap so that the self-interest of each member almost points in the same direction, resulting in a consensus increase.

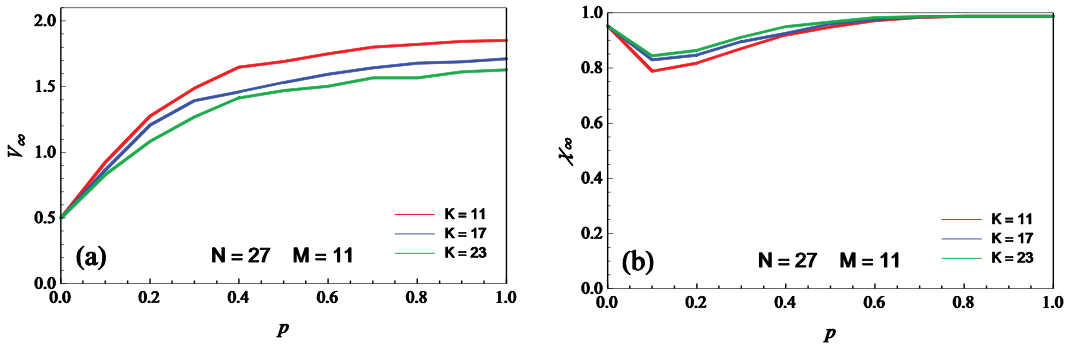


Fig. 6.3, The stationary values of the average group fitness V_∞ and statistically averaged consensus $\chi(t)$, as a function of p. Results are presented for $N = 27$, $M = 11$ and $K = 11, 17, 23$.

Figure 6.4 shows V_∞ and χ_∞ as a function of p for different group sizes $M = 11, 30, 60$ for $N = 27$ and $K = 17$. The same interaction increase parameter μ in eq. (6.3) is used for the team sizes considered. The trend of fitness values with varying p closely follows the ones

already mentioned, with a region of low performance at low p and a region of good performance at high p. For all the group sizes, a rapid transition from low fitness to high fitness values is observed, suggesting the presence of a critical threshold of the control parameter p which triggers the transition from low to high performance of the HGO algorithm.

It is worth noting that the HGO performances [Fig. 6.4 (a)] of the smallest group, $M = 7$, result better of the ones of bigger groups, $M = 30, 60$, especially for very low and high levels of agent's knowledge, and with computational times thousands of times lower [Fig. 6.5]. Nevertheless, the author expects an optimal group size, in terms of performance, can be found for any given optimization problem.

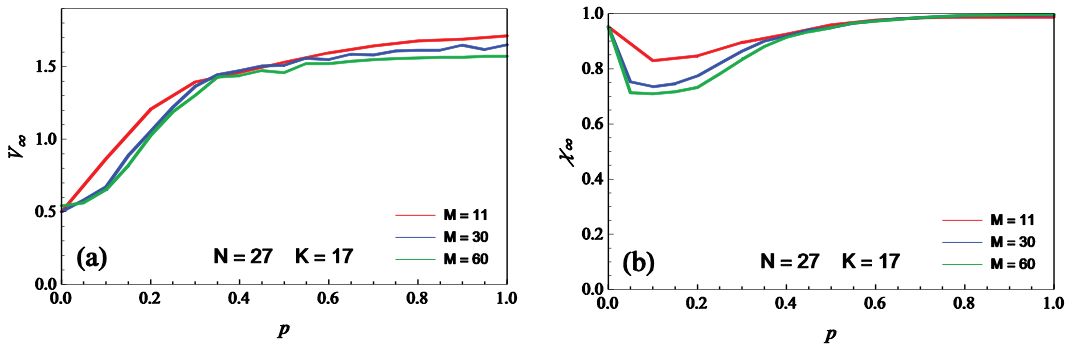


Fig. 6.4, The stationary values of the average group fitness V_∞ and statistically averaged consensus $\chi(t)$, as a function of p. Results are presented for $N = 27, K = 17$ and $M = 11, 30, 60$.

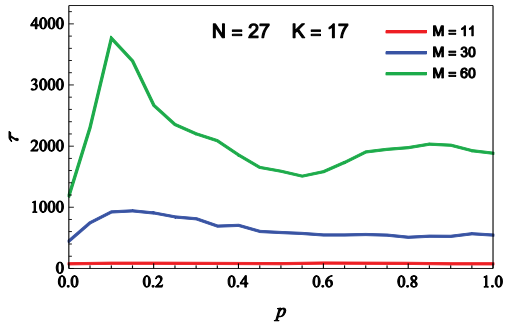


Fig. 6.5, The average time computation τ (seconds) as a function of p, for $M = 11, 30, 60$. Results are showed for $N = 27, K = 17$.

6.4 Comparison with other optimization algorithms

Fig. 6.6 compares the HGO with the MASA, the latter, as before already mentioned, characterized by the absence of social interactions among the agents. Results are shown for $N = 27$, $K = 11, 17, 23$ and $N = 100$, $K = 15$, with p ranging from 0 to 1. For both the algorithms, a group size of $M = 11$ is adopted, with the same values of β_0' previously mentioned.

In all simulations, each stochastic process is simulated by generating 100 different realizations and the ensemble average of the results is calculated. The simulation for HGO is stopped at steady-state, i.e., when changes in the time-averages of consensus and payoff over consecutive time intervals of a given length is sufficiently small. Correspondingly, for MASA, simulations are stopped when the perceived payoff of each agent becomes invariant over successive time intervals.

In all cases, the HGO algorithm outperforms MASA, especially in the case of limited knowledge of the agents. In these situations, the social interaction among the agents pushes them, who do not have knowledge about a certain decision, to make good choice following the decisions of the agents who instead know the influence of the decision on the fitness values, thus making the entire group perform much better compared to the case of non-socially interacting members. We can state that the considerable gap between the red and blue curves in due to the swarm intelligence development in the HGO, absent in MASA.

A comparison of the computational time required by HGO and MASA is reported in Fig. 6.6 (b, d, f, h). It is worth noting that an opposite trend of the curves is shown for the analyzed algorithms. At low knowledge levels, each agent has a limited perception of the real fitness landscape; in these conditions, while MASA agents quickly find inaccurate solutions, HGO agents, trying to reach consensus, share knowledge and information, leading the group to a wiser and clever final set of decisions. Increasing the parameter p, the convergence towards a shared solution in HGO is speeded up, while in MASA each agent, perceiving a bigger search space, requires a longer time for the exploration.

At the intersection of the red and blue curves in Fig. 6.6 (b, d, f, h) HGO and MASA require, on average, the same computational time τ, but evidently, the HGO performances result constantly better of the MASA ones [Fig. 6.6 (a, c, e, g)]. Just in complete knowledge ($p = 1$), the performance of the two algorithms are comparable, and MASA lightly outperforms HGO, requiring, on the other hand, a longer exploration time.

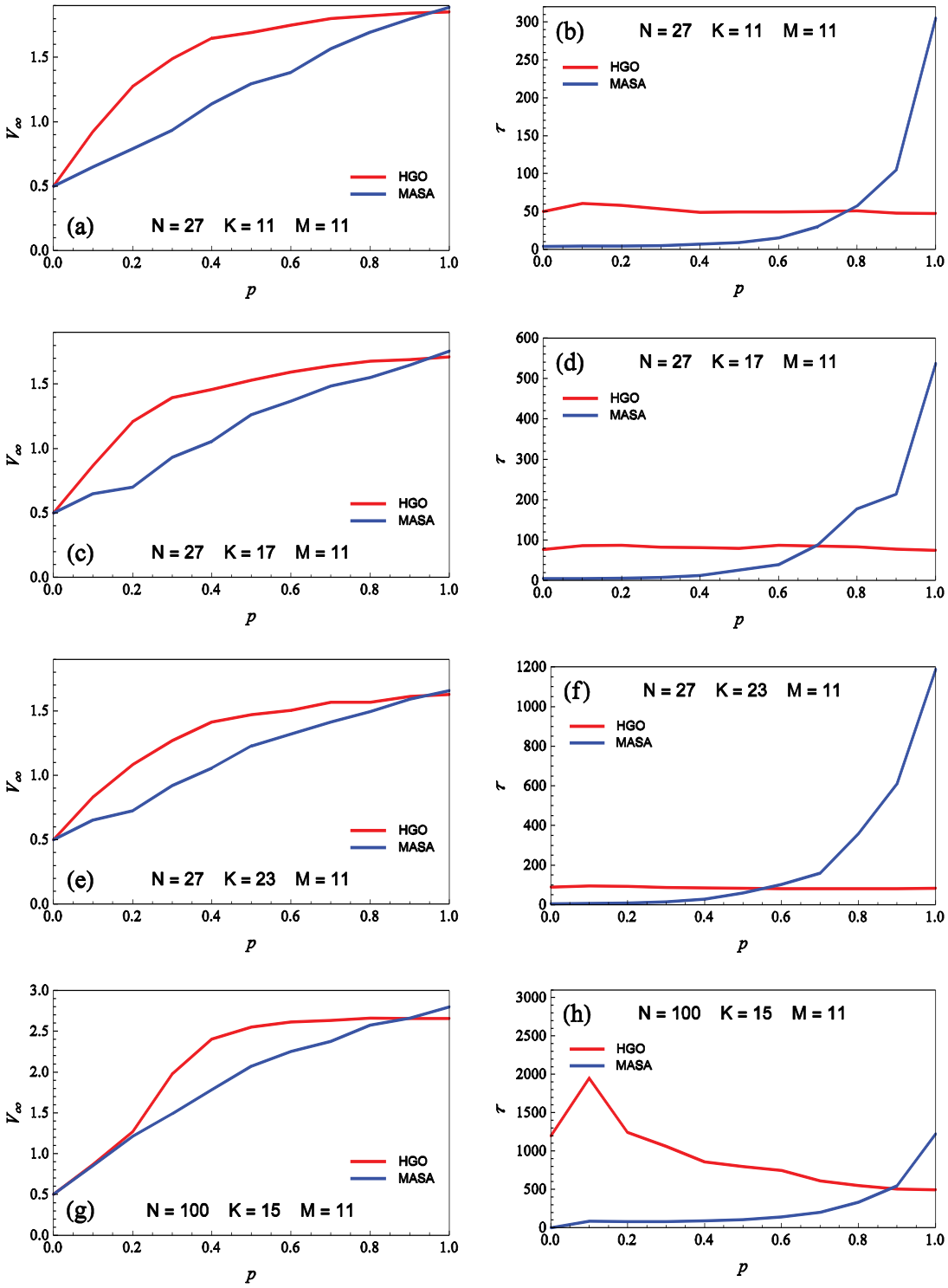


Fig. 6.6, A comparison between HGO and MASA, in terms of steady-state efficacy V_∞ and computational time required τ, as a function of the knowledge level p, for $N = 27, K = 11, 17, 23$, and $N = 100, K = 15$.

In conclusion, we test the efficacy of HGO with those of existing optimization algorithms, particularly suited for solving combinatorial discrete optimization problems, i.e. Simulated Annealing (SA) (see Appendix 8.5) and Genetic Algorithm (GA). For the former, the same

initial temperature values β'^{-1}_0, imposed to HGO and MASA, have been used. About GA, we used the Global Optimization Toolbox of MATLAB© R2015b with 1000 agents, instead of the only 11, used for HGO and MASA.

Table 6.1 shows a comparison, in the case of complete knowledge ($p = 1$), of the four optimization algorithms performance, expressed like commonly happen in optimization, in terms of the averaged best fitness values V_B, i.e. averaging for each simulation the best fitness value found, not necessarily the one at steady-state. The performance found by HGO and MASA are very closed, and always considerably better of the ones of SA e GA.

p=1	**HGO**	**MASA**	**SA**	**GA**
N=27 K= 11	1.993	2.042	1.694	1.565
N=27 K= 17	1.914	1.947	1.679	1.456
N=27 K= 23	1.876	2.017	1.578	1.381
N=100 K= 15	3.005	3.030	1.690	2.420

Tab. 6.1, The comparison of the averaged best fitness values V_B, found by HGO, MASA, SA and GA, for the cases $N = 27, K = 11, 17, 23$, and $N = 100, K = 15$.

Moreover, Fig. 6.7 shows the distribution of the best values found by HGO and MASA for the cases of $N = 27, K = 17$, and $N = 100, K = 15$. They appear to be very narrow around their mean values, compared to SA and GA ones. This is an optimal feature of these algorithms, because it ensures, in some way, that we can trust in the accuracy of each single simulation process, without the needed to run a lot of simulations.

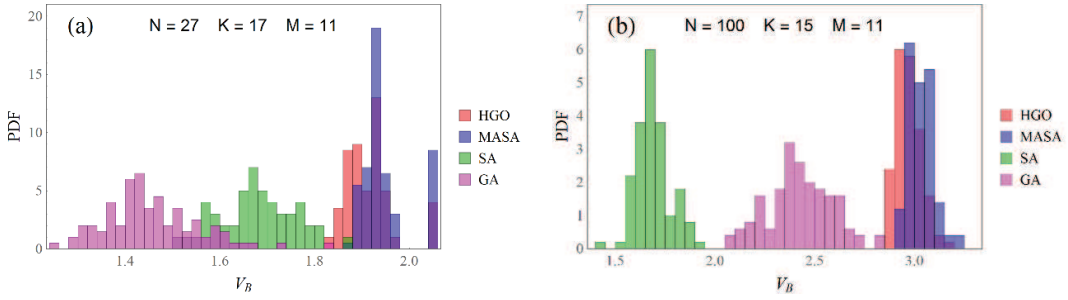


Fig. 6.7, The distribution of the best fitness values, found by HGO, MASA, SA and GA. Results are showed for $N = 27, K = 17$ and $N = 100, K = 15$.

6.5 Conclusions

In this chapter, the author proposed a novel swarm-based optimization algorithm mimicking the collective decision-making behavior of human groups. This algorithm, which we termed Human Group Optimization (HGO), describes the decision process of the agents in terms of a time-continuous Markov chain, where the transition rates are defined so as to capture the effect of the self-interest, which pushes each single agent to increase

the perceived fitness, and of social interactions, which stimulate member to seek consensus with the other members of the group. The Markov chain is, then, characterized by a couple of parameters that, likewise the Simulated Annealing, are subjected to a specific cooling schedule that in the long-time limit makes the system converge in probability to the optimal value. The choice of the parameters is made to guarantee the transition to a consensus state at which the group of agents shows a very high degree of collective intelligence. While being in this state, the agents explore the landscape by sharing information and knowledge through social interactions, so as to achieve very good solutions even in the case of a limited knowledge.

To test the proposed HGO algorithm, we considered the hard-*NP* problem of finding the optimum on *NK* fitness landscape and compared the methodology with other well established algorithms as the Simulated Annealing, a multi-agent version of it and Genetic Algorithm. In all cases, the HGO has been shown to significantly outperform the other two algorithms, especially under limited knowledge conditions.

Summarizing, the algorithm presents several advantages that make it very suitable to solve complex operation management problems. It is flexible because it can be applied to almost any combinatorial problem by identifying the number of decisions the agents should make. However, its most attractive feature relies on its ability to identify very good solutions, even in presence of partial knowledge of the agents. For this reason, it appears very promising for applications in distributed decision-making contexts such as supply chains. Furthermore, while the vast majority of swarm intelligent algorithms, mimicking the behavior of social groups like insects and animals, are based on the mechanism of the stigmergy, our algorithm introduces a mechanism based on the direct communication among individuals, which is a more powerful and effective way to achieve coordination. Under this perspective, the proposed code is novel and unique within the class of swarm intelligent optimization codes.

The algorithm could be also fine-tuned to solve specific operations management problems characterized by distributed decision making and information asymmetry, such as multi-stage production scheduling, location routing problem, supply chain inventory problem, just to name a few.

Of course, this first version of the algorithm could be further improved in future research by identifying better cooling schedules. Additional numerical tests and theoretical investigation are however needed to quantify pros and cons.

7 Conclusions

"Try to learn something about everything
and everything about something."
Thomas Henry Huxley

The purpose of this dissertation is to study the emergence of the collective intelligence of human groups, introducing a new decision-making model (DMM), initially proposed by Carbone and Giannoccaro (Carbone & Giannoccaro, 2015) for solving complex combinatorial problems. Contrarily to other studies proposed in the literature (Bordogna & Albano, 2007), (Turalaska et al., 2009), (Turalaska, West & Grigolini, 2009) based on the mechanism of *imitation*, we recognized that consensus seeking apart, the rational evaluation characterizes human beings, driving humans to compare alternative strategies in terms of costs and benefits and efficiently solve a problem.

The work begins with chapter 2, in which the notion of swarm intelligence in natural systems is introduced, reviewing the mechanisms behind these fascinating behaviors and providing the state of the art in the developing field of swarm robotics.

Following, chapter 3 introduces the concept of optimization problems, revising the most famous biologically inspired optimization algorithms and highlighting their main versions, advantages and drawbacks.

In chapter 4, the DMM is presented, widely analyzing its features and potentialities. The DMM captures the main drivers of the humans' behaviors in groups, i.e., *self-interest* and *consensus seeking*. Agent's choices are made by optimizing the perceived fitness value, which is an estimation of the real one, based on the level of agent's knowledge. Simultaneously, social influence pushes the individual to modify the choice he/she made, for the natural tendency of humans to seek consensus with people they interact with (Di Maggio & Powell, 1983). We found that a moderate strength of social interactions allows for knowledge transfer among the members, leading to a higher knowledge level of the group as a whole. This mechanism, coupled with the ability to explore the fitness landscape, strongly improves the performance of the decision-making process. We also identified that the threshold value of the social interaction strength, at which the entire group is characterized by a higher degree of collective intelligence, is just the critical threshold at

which the flow of information from the fitness landscape to the group of agents is maximized, thus improving the abilities of the group to explore the fitness landscape searching for the optimal solution.

In Chapter 5 the DMM is employed to simulate how a team of individuals, in charge to design a new product, converge towards a shared solution of the design process by interacting with each other. We supposed the interactions among individuals take place for hierarchical or social reasons, investigating respectively the influence of the team hierarchical structure and social network topology on the team performances, and providing indications about how to effectively design a team. It is worth noting that a similar formulation can be easily adapted to any real social decision-making problem.

In chapter 6 a new optimization algorithm, belonging to the class of swarm intelligence optimization methods, is introduced. The algorithm, referred to as Human Group Optimization (HGO), is developed within the previously mentioned DMM (Carbone & Giannoccaro, 2015) and emulates the collective decision-making process of human groups. A continuous-time Markov process is proposed to describe the behavior of a population of socially interacting agents, modeling how humans in a group modify their opinions driven by self-interest and consensus seeking. As for collection of spins, the dynamics of such a system is characterized by a phase transition from low to high values of the consensus (magnetization), associated with the emergence of a collective superior intelligence of the population. While this state being active, a cooling schedule is applied to make agents closer and closer to the optimal solution, while performing their random walk on the fitness landscape. To test the ability of the HGO algorithm, its performance were compared with those of the Simulated Annealing (SA), and Genetic Algorithm (GA) in solving *NP*-complete problems, consisting in finding the optimum on a fitness landscape generated within the Kauffman *NK* model of complexity (Kauffman & Levin, 1987), (Kauffman & Weinberger, 1989), (Weinberger & others, 1996). HGO always outperforms other algorithms, being able to identify very good solutions, even in presence of partial knowledge of the agents. This attractive feature, make it particularly suited for applications in distributed decision-making contexts such as supply chains. It is worth noting the proposed algorithm results flexible because it can be applied to almost any combinatorial problem by identifying the number of decisions the agents should make. Furthermore, differently from the majority of swarm intelligent algorithms, mimicking the mechanism of the stigmergy characterizing the behavior of social groups like insects and animals, HGO introduces a mechanism based

on the direct communication among individuals, which is a more powerful and effective way to achieve coordination. Under this perspective, the proposed code is novel and unique within the class of swarm intelligent optimization codes.

8 Appendix

"If you can't explain it simply
you don't understand it well enough."
Albert Einstein

8.1 The NK fitness landscape

The *NK* model is a mathematical model conceived by Stuart Kauffman (Kauffman & Levin, 1987), (Kauffman & Weinberger, 1989) to generate tunable rugged fitness landscapes. "Tunable ruggedness" captures the intuition that both the overall size of the landscape and the number of its local "hills and valleys" can be adjusted changing its two parameters, *N* and *K*, defined in the following.

Before getting into the detail explanation of the mathematical formulation of the method, the author would like to emphasize the concept of the interdependency of variables in a combinatorial optimization problem and how it effects its difficult.

If variables are independent of each other, optimization is a relatively simple case of figuring out which way to adjust each one of them. However, in most real problems, variables are interdependent, and adjusting one might make another one less effective. Language is an example of this: the meaning of a word depends on its context and the interrelatedness of them is a fundamental issue to well identify their meaning.

Kauffman had the intuition that two factors affect the complexity of a landscape. These factors are N, the size of a problem, and K, the amount of interconnectedness of the elements that make it up. For a better understanding let's consider some simple examples. Consider a network of five binary nodes, with no connections between them [Fig. 8.1 (a)].

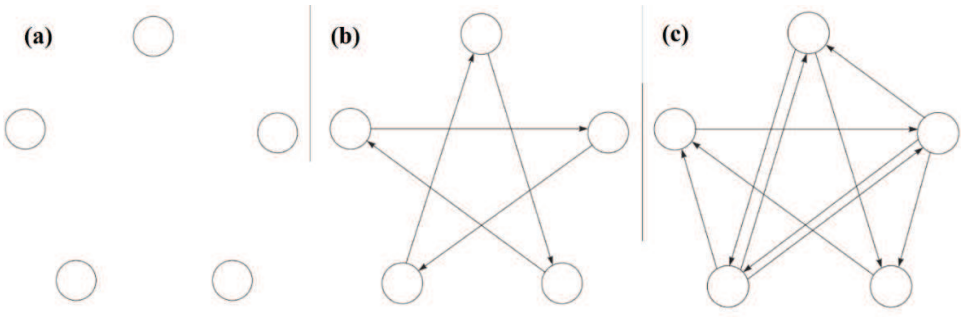


Fig. 8.1, *NK* networks with $N = 5$ and respectively, (a) $K = 0$, (b) $K = 1$, (c) $K = 2$.

In such a trivial network, each node has a preferred value, i.e. it is "better" for it to take on the value of 1 or of -1. In Kauffman's research, a table of fitness values is determined by random numbers. For instance, if node 1 has a state value of 1, its fitness is 0.123, and if it is -1, its fitness is 0.987. We want to maximize fitness, so it is better for this node to have a value of -1. A fitness table for this very simple graph is shown in Table 8.1.

Node	**1**	**2**	**3**	**4**	**5**
Value if -1	0.987	0.333	0.864	0.001	0.789
Value if 1	0.123	0.777	0.923	0.004	0.321

Table 8.1, Example of fitness table for simple *NK* network where $N = 5$ and $K = 0$.

Since these nodes are binary, we can write the state of the network as a bit-string; for instance, $(1, -1, 1, -1, 1)$ means that the first node has a state value of 1, the second of -1, the third of 1, the fourth of -1, and the fifth of 1. Looking at Table 8.1, we note that the best possible state of the network is $(-1, 1, 1, 1, -1)$. This pattern of node states results in the highest fitness at each site in the graph and thus produces the highest sum over the entire graph.

Observe that we are analyzing a very simple case, because while there are five nodes $N = 5$, they don't interact; i.e. none has any effect on the other. Such a system has $K = 0$, where K stands for the average number of inputs that each node receives from other nodes. Obviously, with $K = 0$ it is extremely easy to find the global optimum of the network. We can simply pick a site at random and flip its sign from -1 to 1 or vice versa; if the network's total fitness increases, we leave it in the new state, otherwise we return it to its original state (this is called a *Greedy algorithm*). We only need to perform *N* operations; once we have found the best state for each node.

If we increase K to 1 [Fig. 8.1 (b)], the fitness of a node depends on its own state and the state of the node at the sending end of an arrow pointing to it. In this connected situation, we have to conduct more than *N* operations to find the optimal pattern. It is possible in fact, that reversing the state of a node will increase its performance but decrease the performance of the node it is connected to, or the opposite, its performance will decrease while the other's increases. Further, while we might find a combination that optimizes the pair, the state of the receiving node affects the node it is connected to, too, and it is likely that the state of that node is best when this node is in the opposite state, and so on.

When $K = 2$, the fitness of each node depends on its own state (-1 or 1) and the states of two other nodes whose arrows point to it [Fig. 8.1 (c)]. The size of the lookup table increases exponentially as *K* increases; its size is $2^{K+1}N$. Reversing the state of a node directly affects

the fitness of two other nodes, perhaps changing their optimal state, and their states affect two others, and so on. K can be any number up to $N-1$, at which point every node is connected to every other node in the whole system.

It is simple to understand why Kauffman has theorized that two parameters, N and K, can completely describe the *complexity* of any system.

First, as N, the dimensionality of the system, increases, the number of possible states of the system increases exponentially: remember that there are 2^N arrangements of N binary elements. This increase is known as a *combinatorial explosion*, and even if it seems obvious, this is a significant factor in determining how hard it will be to find an optimal configuration of elements. Each new binary element that is added doubles the patterns of node activations and rapidly becomes impossible to test all possible combinations.

The second factor, K, is also known as *epistasis*. This term has been borrowed from genetics, where it is often seen that the effect of a gene at one site on the chromosome depends on the states of genes at other sites. When $K = 0$, there is only one peak, and as we saw, that peak is found when each node takes on its better value. Kauffman has shown instead that when K becomes higher relative to N, landscapes become irregular and eventually random. In these conditions, the highest peaks are poorer, due to conflicting constraints, and the paths to peaks on the landscape are shorter.

In 1991, Weinberger published a detailed analysis (Weinberger, 1996) of the case in which $1 \ll K \leq N$ and the fitness contributions are chosen randomly. His analytical estimate of the number of local optima was later shown to be flawed. However, numerical experiments included in Weinberger's analysis support his analytical result that the expected fitness of a string is normally distributed with a mean of approximately $\mu + \sigma\sqrt{\frac{2\ln(K+1)}{K+1}}$ and a variance of approximately $\frac{(K+1)\sigma^2}{N[K+1+2(K+2)\ln(K+1)]}$.

Following a three-dimensional representation of *NK* landscapes of increasing complexity.

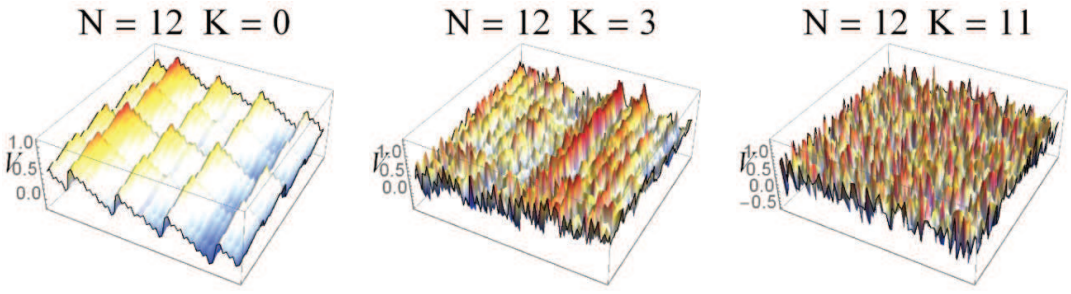


Fig 8.2, *NK* landscapes with $N = 12$ and respectively, (a) $K = 0$, (b) $K = 3$, (c) $K = 11$.

Now let's see how, given a binary N-dimensional decision vector $\boldsymbol{d} = (d_1, d_2, \dots, d_N)$, i.e. $d_i = \pm 1,\ i = 1, 2, \dots, N$, we can calculate the associated payoff $V(\boldsymbol{d})$ following the *NK* procedure. The fitness value $V(\boldsymbol{d})$ is defined as the weighted sum of N stochastic contributions $W_j(d_j, d_1^j, d_2^j, \dots, d_K^j)$, each decision leads to total fitness depending on the value of the decision d_j itself and the values of other K decisions $d_i^j,\ i = 1, 2, \dots, K$. The integer index $K = 0, 1, 2, \dots, N-1$ is the number of interacting decision variables, and tunes the complexity of the problem, that increases with K. For $K > 2$, in computational complexity theory, finding the optimum of the fitness function $V(\boldsymbol{d})$ is classified as an *NP*-complete decision problem (Weinberger, 1996).

All the information required to generate an *NK* Landscape can be stored in two matrixes:

- a **positions matrix**, $\boldsymbol{\Pi}$, with dimensions $N \times (K+1)$, containing for each decision variable (row), the $K+1$ numbers of the interacting decisions variables (columns);
- a **payoff contributions matrix**, $\mathbf{W}$, with dimensions $2^{K+1} \times N$, containing for each decision variable (column), 2^{K+1} random values (rows), drawn from a $[0, 1]$ uniform distribution.

For sake of simplicity let's transform $\boldsymbol{d}$ in a binary string containing only zeros and ones, $\hat{\boldsymbol{d}}$:

$$\hat{\boldsymbol{d}} = \left(\hat{d}_1, \hat{d}_2, \dots, \hat{d}_N\right) = (\boldsymbol{d} + 1)/2 \tag{8.1}$$

For each decision $j \in [1, N]$:

- Consider the *j*th row of $\boldsymbol{\Pi}$, $\boldsymbol{\Pi}_j = \boldsymbol{\Pi}(j, :)$, containing the $K+1$ numbers of the interacting decisions variables with j.
- Select from $\hat{\boldsymbol{d}}$ the binary substring $\boldsymbol{b}^j = \left\{b_1^{(j)}, b_2^{(j)}, \dots, b_{K+1}^{(j)}\right\} = \left\{\hat{d}_i\right\}_{i=\boldsymbol{\Pi}_j}$, and choice the stochastic contribution associated with the decision j from the payoff contributions matrix $\mathbf{W}$, as:

$$W_j\left(d_j, d_1^j, d_2^j, \dots, d_K^j\right) = \mathbf{W}\left(1 + \sum_{l=1}^{K+1} b_l^{(j)} 2^{K+1-l}, j\right) \tag{8.2}$$

The fitness function, associated with the decision vector $\boldsymbol{d}$, is finally defined:

$$V(\boldsymbol{d}) = \frac{1}{N} \sum_{j=1}^{N} W_j(d_j, d_1^j, d_2^j, \dots, d_K^j) \tag{8.3}$$

For a better understanding of the previously steps, let's consider the simple case of $N = 4$ and $K = 2$. Following an example of position and payoff contributions matrixes.

$$\mathbf{\Pi} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 4 \\ 1 & 3 & 4 \\ 1 & 2 & 4 \end{bmatrix}; \mathbf{W} = \begin{bmatrix} 0.0489 & 0.8283 & 0.3831 & 0.7455 \\ 0.4408 & 0.2040 & 0.8969 & 0.8413 \\ 0.8414 & 0.8546 & 0.5645 & 0.2037 \\ 0.7102 & 0.5863 & 0.9893 & 0.7407 \\ 0.8340 & 0.2890 & 0.3806 & 0.7653 \\ 0.4211 & 0.5458 & 0.7955 & 0.1315 \\ 0.8041 & 0.1456 & 0.9789 & 0.2011 \\ 0.5036 & 0.1899 & 0.4764 & 0.0143 \end{bmatrix}$$

Suppose we want to calculate the payoff associated to the decision vector $\boldsymbol{d} = (-1,1,-1,1)$, so that $\widehat{\boldsymbol{d}} = (0,1,0,1)$. Table 8.2 contains for each decision variable the correspondent array $\boldsymbol{b}^j$ and the associated stochastic contribution W_j.

j	$\boldsymbol{b}^j$	W_j
1	(0,1,0)	$\mathbf{W}(3,1) = 0.8414$
2	(0,1,1)	$\mathbf{W}(4,2) = 0.5863$
3	(0,0,1)	$\mathbf{W}(2,3) = 0.8969$
4	(0,1,1)	$\mathbf{W}(4,4) = 0.7407$

Table 8.2, Stochastic contribution associated at each decision j for the decision vector in consideration.

Applying eq. (8.3) it is simple to compute $V(\boldsymbol{d})$:

$$V(\boldsymbol{d}) = \frac{1}{4}(0.8414 + 0.5863 + 0.8969 + 0.7407) = 0.7663$$

The reader can also check the fitness values associated with each decision vector of the small decision space considered. Their values are reported in Table 8.3.

$\boldsymbol{d}$	W_1	W_2	W_3	W_4	$V(\boldsymbol{d})$
$(-1,-1,-1,-1)$	0.0489	0.8283	0.3831	0.7455	***0.5014***
$(-1,-1,-1,+1)$	0.0489	0.204	0.8969	0.8413	***0.4978***
$(-1,-1,+1,-1)$	0.4408	0.8283	0.5645	0.7455	***0.6448***
$(-1,-1,+1,+1)$	0.4408	0.204	0.9893	0.8413	***0.6188***
$(-1,+1,-1,-1)$	0.8414	0.8546	0.3831	0.2037	***0.5707***
$(-1,+1,-1,+1)$	0.8414	0.5863	0.8969	0.7407	***0.7663***
$(-1,+1,+1,-1)$	0.7102	0.8546	0.5645	0.2037	***0.5832***
$(-1,+1,+1,+1)$	0.7102	0.5863	0.9893	0.7407	***0.7566***
$(+1,-1,-1,-1)$	0.834	0.289	0.3806	0.7653	***0.5672***
$(+1,-1,-1,+1)$	0.834	0.5458	0.7955	0.1315	***0.5767***
$(+1,-1,+1,-1)$	0.4211	0.289	0.9789	0.7653	***0.6138***
$(+1,-1,+1,+1)$	0.4211	0.5458	0.4764	0.1315	***0.3937***
$(+1,+1,-1,-1)$	0.8041	0.1456	0.3806	0.2011	***0.3828***
$(+1,+1,-1,+1)$	0.8041	0.1899	0.7955	0.0143	***0.4509***
$(+1,+1,+1,-1)$	0.5036	0.1456	0.9789	0.2011	***0.4573***
$(+1,+1,+1,+1)$	0.0489	0.8283	0.3831	0.7455	***0.2960***

Table 8.3, Decision vectors, stochastic contributions and fitness values, associated to the $N = 4$, $K = 2$ Landscape considered as example.

8.2 The Glauber dynamics of Ising model on general graphs

The statistical study of systems of interacting particles is affected by many problems, largely mathematical in nature. Theorists devoted a great effort to devising and studying the simplest sorts of model systems which show any resemblance to those occurring in nature. One of the most successful models was introduced by Ernst Ising (Ising, 1925) in an attempt to explain the ferromagnetic phase transition and completely solved for the first time by Onsager (Onsager, 1944) for the two-dimensional case.

In the Ising model, a general graph is populated by n spins that may take one of two values: $s_i = \pm 1$. The condition of the system can be defined by the state vector $\boldsymbol{s} = (s_1, s_2, \dots, s_n)$. Defined the adjacency matrix $\boldsymbol{A} = A_{ij}$, the energy of the system can be expressed as:

$$E(\boldsymbol{s}) = -\frac{1}{2}\frac{J}{\langle k \rangle}\boldsymbol{A}\boldsymbol{s} \cdot \boldsymbol{s} = -\frac{1}{2}\frac{J}{\langle k \rangle}\sum_{ij} A_{ij} s_i s_j \tag{8.4}$$

Observe that $A_{ii} = 0$ (with $i = 1, \dots, n$). In equation (8.4) the term 1/2 avoids that each couple of spins *i* and *j* be double counted, and $\langle k \rangle$ is the mean degree of the network.

Pairs of nearest neighbor spins experience a ferromagnetic interaction that favors their alignment. Every parallel pair of neighboring spins contributes $-J/\langle k \rangle$ to the energy and every antiparallel pair contributes $+J/\langle k \rangle$. When the coupling constant is positive, the interaction favors ferromagnetic order. The main feature of the Ising model is that ferromagnetism appears spontaneously in the absence of any driving field when the temperature T is less than a critical temperature T_c and the spatial dimension $d > 1$. Above T_c, the spatial arrangement of spins is spatially disordered, with equal numbers of spins in the states +1 and -1. Consequently, the magnetization, $\langle s \rangle = n^{-1} \sum_i s_i$, is zero and spatial correlations between spins decay exponentially with their separation. Below T_c, the magnetization is non-zero and distant spins are strongly correlated.

All thermodynamic equilibrium properties of the Ising model can be obtained from the partition function $Z = \sum_l \exp[-\beta E(\boldsymbol{s}_l)]$, where the sum is over all spin configurations of the system, with $\beta = 1/k_B T$ and k_B is the Boltzmann constant.

Contrarily, the non-equilibrium properties depend on the nature of the spin dynamics. There is considerable freedom in formulating these dynamics that is dictated by physical considerations. For example, the spins may change one at a time or in correlated blocks. More fundamentally, the dynamics may or may not conserve the magnetization. This lack

of uniqueness of dynamical rules is the reason why there do not exist universal principles that prescribe how to solve a non-equilibrium spin system.

In 1963 Glauber (Glauber, 1963) proposed a non-conservative single-spin-flip dynamic, in which spins are selected one at a time in random order and each one changes at a rate that depends on the change in the energy of the system as a result of this update. Because only single spins can change sign in an update, $s_l \rightarrow -s_l$, where s_l is the spin value at site l, the magnetization is generally not conserved.

Three types of transitions can arise when single spin flips: *energy raising*, *energy lowering*, and *energy neutral transitions* [Fig. 8.3]. Energy raising events occur when a spin is aligned with a majority of its neighbors and vice versa for energy lower events. Energy conserving events occur when the net magnetization of the neighbors is zero.

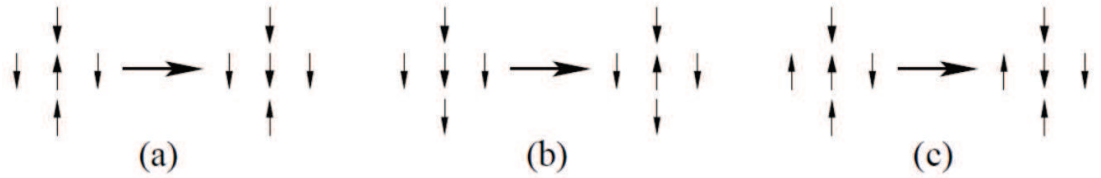


Fig. 8.3, (a) Energy lowering, (b) energy raising, and (c) energy conserving spin-flip events on the square lattice.

The basic principle to fix the rates of the various types of events is the *detailed balance condition*, mathematically expressed by:

$$\frac{w(\boldsymbol{s}_l \rightarrow \boldsymbol{s}_l')}{w(\boldsymbol{s}_l' \rightarrow \boldsymbol{s}_l)} = \frac{P_0(\boldsymbol{s}_l')}{P_0(\boldsymbol{s}_l)} \tag{8.5}$$

Here $\boldsymbol{s}_l = (s_1, s_2, \dots, s_l, \dots, s_n)$ denotes the state of all the spins in the system, $\boldsymbol{s}_l' = (s_1, s_2, \dots, -s_l, \dots, s_n)$ denotes the state derived from $\boldsymbol{s}_l$ in which the spin s_l flipped to $-s_l$, and $w(\boldsymbol{s}_l \rightarrow \boldsymbol{s}_l')$ represents the transition rate from $\boldsymbol{s}_l$ to $\boldsymbol{s}_l'$.

In steady state conditions the stationary probability distribution of the states, $P_0(\boldsymbol{s})$, is given by the Boltzmann distribution:

$$P_0(\boldsymbol{s}) = \frac{\exp[-\beta E(\boldsymbol{s})]}{Z} \tag{8.6}$$

The meaning of the detailed balance condition is simple. In the abstract space of all 2^n possible spin states of a system of n spins, Glauber dynamics connects states which differ by the reversal of a single spin. When detailed balance holds, the probability currents from state $\boldsymbol{s}_l$ to $\boldsymbol{s}_l'$ and from $\boldsymbol{s}_l'$ to $\boldsymbol{s}_l$ are equal so there is no net probability current across any link in this state space.

Now observe that:

$$E(\boldsymbol{s}_l) = -\frac{J}{\langle k\rangle} s_l \sum_j A_{lj} s_j - \frac{1}{2}\frac{J}{\langle k\rangle}\sum_{ij\neq l} A_{ij} s_i s_j \tag{8.7}$$

Substituting equations (8.6) and (8.7) in equation (8.5):

$$\frac{w(\boldsymbol{s}_l \to \boldsymbol{s}_l')}{w(\boldsymbol{s}_l' \to \boldsymbol{s}_l)} = \frac{P_0(\boldsymbol{s}_l')}{P_0(\boldsymbol{s}_l)} = \frac{\exp\left[-\beta\frac{J}{\langle k\rangle} s_l \sum_j A_{lj} s_j + \frac{1}{2}\beta\frac{J}{\langle k\rangle}\sum_{ij\neq l} A_{ij} s_i s_j\right]}{\exp\left[\beta\frac{J}{\langle k\rangle} s_l \sum_j A_{lj} s_j + \frac{1}{2}\beta\frac{J}{\langle k\rangle}\sum_{ij\neq l} A_{ij} s_i s_j\right]}$$

$$\frac{w(\boldsymbol{s}_l \to \boldsymbol{s}_l')}{w(\boldsymbol{s}_l' \to \boldsymbol{s}_l)} = \frac{\exp\left[-\beta\frac{J}{\langle k\rangle} s_l \sum_j A_{lj} s_j\right]}{\exp\left[\beta\frac{J}{\langle k\rangle} s_l \sum_j A_{lj} s_j\right]}$$

Recalling that $\exp(x) = \cosh(x)\,[1 + \tanh(x)]$, and that $\cosh(x)$ and $\tanh(x)$ are respectively, even $[\cosh(-x) = \cosh(x)]$ and odd $[\tanh(-x) = -\tanh(x)]$ functions, we get:

$$\frac{w(\boldsymbol{s}_l \to \boldsymbol{s}_l')}{w(\boldsymbol{s}_l' \to \boldsymbol{s}_l)} = \frac{1 - \tanh\left(\beta\frac{J}{\langle k\rangle} s_l \sum_j A_{lj} s_j\right)}{1 + \tanh\left(\beta\frac{J}{\langle k\rangle} s_l \sum_j A_{lj} s_j\right)}$$

Observe that $s_l = \pm 1$, so that

$$\tanh\left(\beta\frac{J}{\langle k\rangle} s_l \sum_j A_{lj} s_j\right) = s_l \tanh\left(\beta\frac{J}{\langle k\rangle}\sum_j A_{lj} s_j\right)$$

and we finally obtain:

$$\frac{w(\boldsymbol{s}_l \to \boldsymbol{s}_l')}{w(\boldsymbol{s}_l' \to \boldsymbol{s}_l)} = \frac{1 - s_l \tanh\left(\beta\frac{J}{\langle k\rangle} \sum_j A_{lj} s_j\right)}{1 + s_l \tanh\left(\beta\frac{J}{\langle k\rangle} \sum_j A_{lj} s_j\right)} \tag{8.8}$$

Therefore, a possible choice for the transition rates for the Ising-Glauber dynamics on general graph is:

$$w(\boldsymbol{s}_l \to \boldsymbol{s}_l') = \alpha\left[1 - s_l \tanh\left(\beta\frac{J}{\langle k\rangle}\sum_j A_{lj} s_j\right)\right] \tag{8.9}$$

where α is an arbitrary constant. In our model, we have chosen $\alpha = 1/2$. This transition rate ensures that any initial spin state will eventually relax to the equilibrium thermodynamic equilibrium state for any non-zero temperature.

8.3 Detailed balance condition

In this brief paragraph, we show that the transition rate proposed in equation (4.6) fulfills the detailed balance condition of Markov chains, which requires the existence of a stationary probability distribution $P_0(\boldsymbol{s}_l)$ such that

$$\frac{P_0(\boldsymbol{s}_l')}{P_0(\boldsymbol{s}_l)} = \frac{w(\boldsymbol{s}_l \to \boldsymbol{s}_l')}{w(\boldsymbol{s}_l' \to \boldsymbol{s}_l)} \tag{8.10}$$

Using equation (4.6) the above condition equation (8.10) writes

$$\frac{P_0(\boldsymbol{s}_l')}{P_0(\boldsymbol{s}_l)} = \frac{1 - s_l \tanh\left(\beta \frac{J}{\langle k \rangle} \sum_j A_{lj} s_j\right)}{1 + s_l \tanh\left(\beta \frac{J}{\langle k \rangle} \sum_j A_{lj} s_j\right)} \frac{\exp\{\beta'[\bar{V}(\boldsymbol{s}_l') - \bar{V}(\boldsymbol{s}_l)]\}}{\exp\{\beta'[\bar{V}(\boldsymbol{s}_l) - \bar{V}(\boldsymbol{s}_l')]\}} \tag{8.11}$$

and recalling equations (8.5), (8.6) and (8.9) yields

$$\frac{P_0(\boldsymbol{s}_l')}{P_0(\boldsymbol{s}_l)} = \frac{\exp[-\beta E(\boldsymbol{s}_l') + 2\beta' \bar{V}(\boldsymbol{s}_l')]}{\exp[-\beta E(\boldsymbol{s}_l) + 2\beta' \bar{V}(\boldsymbol{s}_l)]} \tag{8.12}$$

This allows defining the stationary probability distribution

$$P_0(\boldsymbol{s}_l) = \frac{\exp[-\beta E(\boldsymbol{s}_l) + 2\beta' \bar{V}(\boldsymbol{s}_l)]}{\sum_k \exp[-\beta E(\boldsymbol{s}_k) + 2\beta' \bar{V}(\boldsymbol{s}_k)]} \tag{8.13}$$

which satisfies the detailed balance condition equation (8.10).

8.4 The exponential distribution of events and the Gillespie Algorithm

In chapter 4 we introduced the decision-making model (DMM), highlighting that the process we deal with is a homogenous continues-time Markov chain. This means that the evolution of the system depends only on its current state and not on its previous history; similarly, the transition rates depend only on the actual state and not explicitly by time.

Let be $\boldsymbol{s}(t_0)$ the state vector of the system, containing the opinion of the M agents on the N decisions, at time t_0. $\boldsymbol{s}(t_0)$ can evolve in $n = N{\times}M$ different state vectors, $\boldsymbol{s}_l$, each one equal to $\boldsymbol{s}$, but with the *l*th opinion changed from s_l to $-s_l$. Let us define the probability $\Psi_l(t)$ that one of the possible events E_l: $(\boldsymbol{s} \to \boldsymbol{s}_l)$ occurs in the time interval $]t_0, t_0 + t]$, where t is an arbitrary waiting time, starting from the observation time t_0.

For the *memoryless property* of Markov chains the observation time t_0 does not affect the probability that an event occurs, which only depends on the waiting time t, such that we can always reset $t_0 = 0$ at each new starting observation. Also, we can realistically assume that the probability that each one of the possible events, i.e. each one of the possible opinion changes, occurs in the time interval $]t, t + dt]$ is proportional to the length of that time interval dt, through a constant transition rate w_l.

$$\Psi_l\big(t < t_{E_l} \leq t + dt\big) = w_l\, dt \tag{8.14}$$

The probability $d\Psi_l(t)$ that the opinion change E_l occurs in the time interval $]t, t + dt]$ and not before, is the joint probability that E_l does not occur up to time t, $[1 - \Psi(t)]$, and that it occurs in the time interval $]t, t + dt]$, $w_l\, dt$. Due to the independence of this events, the joint probability is equal to the product of the marginals.

$$d\Psi_l(t) = [1 - \Psi_l(t)]w_l\, dt \tag{8.15}$$

Rearranging eq. (8.15) we get a first order constant coefficients linear differential equation.

$$\frac{d\Psi_l(t)}{dt} + w_l\Psi_l(t) = w_l \tag{8.16}$$

Solving the latter and imposing the initial condition $\Psi_l(0) = 0$, we find an exponential distribution of the time events:

$$\Psi_l(t) = 1 - e^{-w_l t} \tag{8.17}$$

and its associated probability density function:

$$\psi_l(t) = \frac{d\Psi_l(t)}{dt} = w_l e^{-w_l t} \tag{8.18}$$

Observe that the expected value of an exponential random variable is $1/w_l$.

The time evolution of the probability $P(\boldsymbol{s}, t)$ obeys the following master equation:

$$\frac{dP(\boldsymbol{s},t)}{dt} = -\sum_{l} w(\boldsymbol{s}_l \to \boldsymbol{s}'_l)P(\boldsymbol{s}_l,t) + \sum_{l} w(\boldsymbol{s}'_l \to \boldsymbol{s}_l)P(\boldsymbol{s}'_l,t) \qquad (8.19)$$

where $\boldsymbol{s}_l = (s_1, s_2, \dots, s_l, \dots, s_n)$, $\boldsymbol{s}'_l = (s_1, s_2, \dots, -s_l, \dots, s_n)$ and the transition rate $w(\boldsymbol{s}_l \to \boldsymbol{s}'_l)$ is the probability per unit time that the opinion s_l flips to $-s_l$ while the others remain temporarily fixed. To calculate the stochastic time evolution of the system, it is possible to follow two different approaches. The first one is the integration of the master equation (8.19), a procedure that requires knowing $P(\boldsymbol{s},t)$ from the beginning of the process and that, although both exact and elegant, results usually not very useful and computational acceptable for making practical calculation, considering that the size of the system scales exponentially (2^n) with the total number of opinions $(n = N{\times}M)$. The second approach consists in using of stochastic simulation algorithms, avoiding to deal with the master equation directly. One of most famous stochastic simulation algorithm was proposed by Gillespie in 1976 (Gillespie, 1976), (Gillespie D. T., 1977) to simulate the time evolution of the stochastic formulation of chemical kinetics, and currently, it is widely employed to simulate continues time Markov Chains models. Two mathematically equivalent procedures were originally proposed by him, the "Direct method" and the "First Reaction method". Both procedures are exact and rigorously based on chemical master equations; however, the direct method is the one typically implemented due to its efficiency and the one we use and we are going to illustrate in this paragraph.

A state vector $\boldsymbol{s}(t_0) = (s_1, s_2, \dots, s_l, \dots, s_n)$, describe the opinions of the M agents regarding the N decisions at a specific time $t_0 = 0$ (remember we can always reset the observation time t_0 to 0, due to the memoryless property of Markov chains).

To move the system forward in time we need to answer two questions:

1. *When will the next opinion-change occur?*
2. *Which agent and which his/her opinion will change?*

1° STEP. The actual state vector $\boldsymbol{s}$ at time $t_0 = 0$ can evolve in $n = N{\times}M$ different state vectors, $\boldsymbol{s}_l$, each one equal to $\boldsymbol{s}$, but with the *l*th opinion changed from s_l to $-s_l$, i.e. $\boldsymbol{s}_1 = (-s_1, s_2, \dots, s_l, \dots, s_n)$, $\boldsymbol{s}_2 = (s_1, -s_2, \dots, s_l, \dots, s_n)$, … , $\boldsymbol{s}_n = (s_1, s_2, \dots, s_l, \dots, -s_n)$, and we are interested in calculating the time delay τ, starting from t_0, after that $\boldsymbol{s}$ will evolve in one of the $\boldsymbol{s}_l$, whatever it will be.

Since all the possible states are independent, the probability $d\Psi$ that one of them will occur in the time interval $[t, t+dt[$ is the sum of the probabilities that each one of them $d\Phi_l$ will happen in the same time interval and not others before.

$$d\Phi = \sum_{l=1}^{n} d\Phi_l \tag{8.20}$$

Due to the independence of these events we can expand each $d\Phi_l$:

$$d\Phi = d\Psi_1[1-\Psi_2(t)]\dots[1-\Psi_n(t)] + [1-\Psi_1(t)]d\Psi_2\dots[1-\Psi_n(t)] + \cdots$$
$$+ [1-\Psi_1(t)][1-\Psi_2(t)]\dots[1-\Psi_{n-1}(t)]d\Psi_n$$

where $d\Psi_1$ is the probability that the event E_1: $(\boldsymbol{s} \to \boldsymbol{s}_1)$ occur in the time interval $]t, t+dt]$, while $[1-\Psi_2(t)]$ is the probability that the event E_2: $(\boldsymbol{s} \to \boldsymbol{s}_2)$ does not occur up to time t, and so on. Introducing the probability densities ψ_l in $d\Psi_l = dt\ \psi_l = dt\ w_l e^{-w_l t}$, we get:

$$d\Phi = dt\ \psi_1\ [1-\Psi_2(t)]\dots[1-\Psi_n(t)] + [1-\Psi_1(t)]dt\ d\psi_2\dots[1-\Psi_n(t)] + \cdots$$
$$+ [1-\Psi_1(t)][1-\Psi_2(t)]\dots[1-\Psi_{n-1}(t)]dt\ \psi_n$$

$$d\Phi = dt\ \{w_1 e^{-w_1 t}\ [1-(1-e^{-w_2 t})]\dots[1-(1-e^{-w_n t})] + \cdots$$
$$+ [1-(1-e^{-w_1 t})][1-(1-e^{-w_2 t})]\dots[1-(1-e^{-w_{n-1} t})]w_n e^{-w_n t}\}$$

$$d\Phi = dt\ \{w_1 e^{-w_1 t}\ e^{-w_2 t}\dots e^{-w_n t} + w_2 e^{-w_1 t}\ e^{-w_2 t}\dots e^{-w_n t} + \cdots$$
$$+ w_n e^{-w_1 t}\ e^{-w_2 t}\dots e^{-w_n t}\}$$

$$d\Phi = dt\ (w_1 + w_2 + \cdots + w_n)e^{-(w_1+w_2+\cdots+w_n)t}$$

Defining $w_T = \sum_{l=1}^{n} w_l$, we discover that the probability $d\Phi$ that one of the possible opinion changes occur in the time interval $]t, t+dt]$ is still an exponential with a transition rate, w_T, equals to the sum of all the rates w_l.

$$d\Phi = dt\ w_T e^{-w_T t} \tag{8.21}$$

At this point, we can calculate the time τ to the next opinion flip drawing from an exponential distribution with mean $1/w_T$. For this purpose, it is helpful to spend some words about the problem of drawing numbers from an assigned probability distribution. It is known in fact that calculators are just able to draw numbers from a $[0,1]$ uniform distribution. In the case of our interest, we want to find a relation $y = g(\tau)$, such that drawing τ from an exponential distribution, y results to be $[0,1]$ uniform distributed. Supposing $g(\tau)$ monotone and bijective, the following probability equality should be respected:

$$p(y)dy = p[g(\tau)]d\tau \tag{8.22}$$

Let's call for sake of simplicity $p(\tau) = p[g(\tau)]$. As explained, $p(y) = c$, where c is a constant, while $p(\tau) = w_T e^{-w_T \tau} d\tau$. Substituting in equation (8.22)

$$c\ dy = w_T e^{-w_T \tau} d\tau$$

and integrating the previous equation, we get:

$$cy = 1 - e^{-w_T \tau} + a$$

Imposing the normalization condition $\int_0^1 p(y) = 1$ and the stationary solution $y(t \to +\infty) = 1$, it is simple to calculate the values of the two constants, $c = 1$ and $a = 0$, getting finally $y = 1 - e^{-w_T \tau}$ whence $\tau = -{w_T}^{-1} \ln(1 - y)$. If y is uniformly distributed in $[0, 1]$ also $r = 1 - y$ will be, so that choosing a real random number $0 \leq r < 1$ from a uniform distribution and set $\tau = -w_T^{-1} \log(r)$, τ will result exponentially distributed.

2° STEP. In the second step of the Gillespie algorithm, we should identify which of the possible events E_l: $(\boldsymbol{s} \to \boldsymbol{s}_l)$ occurs. Let's call simply E the event *"the system goes in one of the n possible states $\boldsymbol{s}_l$, starting from $\boldsymbol{s}$"*; the conditional probability that the generic event E_l occurs, given E, is:

$$P(E_l|E) = \frac{P(E_l, E)}{P(E)} = \frac{P(E_l)}{P(E)} = \frac{d\Phi_l}{d\Phi} = \frac{dt\, w_l e^{-w_T t}}{dt\, w_T e^{-w_T t}} = \frac{w_l}{w_T} \tag{8.23}$$

Considering a container containing n groups of spheres, each one with a number of spheres proportional to the ratio w_l/w_T, we can just draw a sphere and consider the correspondent event as happened. This operation is equivalent to normalize the transition rates as $\nu_l = w(\boldsymbol{s}_l \to \boldsymbol{s}_l')/w_T$, to construct the cumulative distribution $F(\nu_l)$ from the probability mass function ν_l, and identify the kth opinion s_k which flips from s_k to $-s_k$ by drawing from a discrete distribution with probability mass function ν_l, i.e. draw a real random number $0 \leq q < 1$ from a uniform distribution and choose k so that $F(\nu_{k-1}) \leq q < F(\nu_k)$.

In the following we just summarize the main steps of the algorithm:

1. Choose a random initial state $\boldsymbol{s}$ of the system.
2. Calculate the transition rates $w(\boldsymbol{s}_l \to \boldsymbol{s}_l')$.
3. Calculate the total rate $w_T = \sum_{l=1}^{n} w_l$.
4. Normalize the transition rates as $\nu_l = w(\boldsymbol{s}_l \to \boldsymbol{s}_l')/w_T$.
5. Construct the cumulative distribution $F(\nu_l)$ from the probability mass function ν_l.
6. Calculate the time τ to the next opinion flip drawing from an exponential distribution with mean $1/w_T$, i.e. choose a real random number $0 \leq r < 1$ from a uniform distribution and set $\tau = -w_T^{-1} \log(r)$.
7. Identify kth opinion s_k which flips from s_k to $-s_k$ by drawing from a discrete distribution with probability mass function ν_l, i.e. draw a real random number $0 \leq q < 1$ from a uniform distribution and choose k so that $F(\nu_{k-1}) \leq q < F(\nu_k)$.
8. Update the state vector and return to step 2 or quit.

8.5 Simulated Annealing and Optimal Initial Temperature

The Simulating Annealing (SA) algorithm for solving combinatorial optimization problems was independently introduced in the early 1980s by Kirkpatrick et al. (Kirkpatrick, Gelatt Jr, & Vecchi, 1983) and Černý (Černý, 1985).

The concept behind the method is inspired by the physical annealing process of solids, that consists in increase the temperature of the heat bath to a maximum value, at which the solid melts, and decrease carefully it until the particles arrange themselves in the ground state of the solid. If the maximum value of the temperature is adequately high and the cooling is performed sufficiently slowly, the particles arrange themselves in a highly-structured lattice, for which the corresponding energy is minimal, otherwise, the solid will be frozen into a meta-stable state.

In 1953 Metropolis, et al. (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953) introduced a simple algorithm, based on Monte Carlo techniques (Binder, 1978), for simulating the evolution of a solid in a heat bath to thermal equilibrium: given a current state i of the solid with energy E_i, a new state j with energy E_j, is generated by applying a perturbation mechanism. If the energy difference, $E_j - E_i$, is less than or equal to zero, the state j is accepted as the current state, otherwise it is accepted with a probability $\exp\left(-\frac{E_j - E_i}{k_B T}\right)$, where T and k_B are respectively the temperature of the heat bath and the Boltzmann constant.

If the lowering of the temperature in the Metropolis algorithm is done sufficiently slowly (a large number of transitions at a given value of the temperature) the solid can reach thermal equilibrium at each temperature. Thermal equilibrium is characterized by the Boltzmann distribution, which gives the probability of the solid of being in a state $\boldsymbol{s}_i$ with energy E_i at temperature T:

$$P(\boldsymbol{s}_i) = \frac{\exp(-E_i/k_B T)}{\sum_j \exp(-E_j/k_B T)} \tag{8.24}$$

The summation extends over all possible states.

Returning to SA, the Metropolis algorithm can be used to generate a sequence of solutions of a combinatorial optimization problem assuming the equivalence between problem solutions and states of the physical system, and between the cost of a solution and the energy of a state. Essentially, SA can be viewed as an iteration of Metropolis algorithms, executed at decreasing values of a control parameter, which play the same role of the temperature, and

that we will call T for analogy. Let's assume we are dealing with a minimization problem; the discussion easily translates to maximization problems. The following pseudo-code summarize the main steps of SA for a combinatorial optimization problem: INITIALIZE computes a start solution and initial values of the parameters T and L (number of iterations at temperature T); GENERATE selects a solution from the neighborhood of the current solution; CALCULATE LENGTH and CALCULATE CONTROL compute new values for the parameters L and T, respectively.

---

procedure SIMULATED ANNEALING;
begin
  INITIALIZE (i_{start}, T_o, L_o);
  $k := 0$;
  $i := i_{start}$;
  repeat
    for $l := 1$ **to** L_k
    begin
      GENERATE (j from S_i);
      if $E(j) \leq E(i)$ **then** $i := j$
      else
      if $\exp\left[-\frac{E(j)-E(i)}{T_k}\right] > random[0,1[$ **then** $i := j$
    end;
    $k = k + 1$;
    CALCULATE LENGTH (L_k);
    CALCULATE CONTROL (T_k);
until stopcriterion
end;

Fig. 8.4, The Simulated Annealing algorithm in pseudo-code.

A distinctive feature of simulated annealing is that, in addition to accepting improvements in cost, it also accepts deteriorations, proportionally to the value of T, i.e. large deteriorations at large values of T and small ones at small values of T, until as T approaches 0, no deteriorations will be accepted at all. This feature means that simulated annealing can escape from local minima, in contrast for example to *iterative improvement* (SA with $T = 0$). The speed of convergence of the algorithm is determined by the choice of the parameters L_k and T_k with $k = 0, 1, ...$, where L_k and T_k denote the values of L and k in iteration k of the algorithm. It has been proved (Aarts, Korst, & Michiels, 2014) that under certain mild conditions on the choice of the parameters, SA converges asymptotically to globally optimal solutions.

It is worth nothing that SA can be also modeled by means of Markov chains (Feller, 1950), (Isaacson & Madsen, 1976), (Seneta, 1981). For each value of the control parameter T_k, a sequence of Markov chains is generated. Each chain consists of a sequence of trials, where the outcomes of the trials correspond to solutions of the problem instance.

Let (Ω, E) be a minimization problem instance, Γ a neighborhood function, and $\boldsymbol{X}(k)$ a stochastic variable denoting the outcome of the *k*th trial. The transition probability at the *k*th trial for each pair $i, j \in \Omega$ of outcomes is defined as:

$$P_{ij}(k) = \mathbb{P}\{\boldsymbol{X}(k) = j | \boldsymbol{X}(k-1) = i\} = \begin{cases} G_{ij}(T_k)A_{ij}(T_k) & \text{if } i \neq j \\ 1 - \sum_{l \in \Gamma, l \neq i} G_{il}(T_k)A_{il}(T_k) & \text{if } i = j \end{cases} \quad (8.25)$$

where $G_{ij}(T_k)$ denotes the *generation* probability, i.e. the probability of generating a solution j when being at solution i, and $A_{ij}(T_k)$ denotes the *acceptance* probability, i.e. the probability of accepting solution j, once it is generated from solution i. The most frequently used choice for these probabilities is the following (Aarts & Korst, 1989):

$$G_{ij}(T_k) = \begin{cases} |\Gamma(i)|^{-1} & \text{if } j \in \Gamma(i) \\ 0 & \text{if } j \notin \Gamma(i) \end{cases} \quad (8.26)$$

and

$$A_{ij}(T_k) = \begin{cases} 1 & \text{if } E(j) \leq E(i) \\ \exp\{-[E(j) - E(i)]/T_k\} & \text{if } E(j) > E(i) \end{cases} \quad (8.27)$$

where $\Gamma(i)$ is the set of neighbors of i. The resulting stationary distribution is then given by:

$$\pi_i = \frac{|\Gamma(i)| \exp(-E_i/T)}{\sum_j |\Gamma(j)| \exp(-E_j/T)} \quad (8.28)$$

A finite-time implementation of simulated annealing is obtained by generating a sequence of homogeneous Markov chains of finite length at descending values of the control parameter T. For this scope, a set of parameters, defining the so-called *cooling schedule*, must be specified:

1. an *initial value* of the control parameter T_o;
2. a *decrement function* $T(k)$ for lowering the value of the control parameter;
3. a *final value* of the control parameter specified by a *stop criterion*;
4. a finite *length* L_k of each homogeneous Markov chain.

The search for adequate cooling schedules has been the subject of many studies over the past years. Reviews are given by (Aarts E. E., 1985) and (Romeo & Sangiovanni-Vincentelli, 1991). One of the most used cooling schedules was proposed by Kirkpatrick (Kirkpatrick,

Gelatt Jr, & Vecchi, 1983) and it is known as *geometrical schedule*, summarized in the following.

1. *Initial value* of the control parameter: $T_o = \Delta E_{max}$, where ΔE_{max} is the maximal difference in cost between any two neighboring solutions. Exact calculation of ΔE_{max} is quite time consuming in many cases, but one can at least give estimates of its value.
2. *Decrement function*: $T_k = \alpha T_{k-1}$, for $k = 0, 1, ...$, where α is a positive constant smaller than but close to 1. Typical values lie between 0.8 and 0.99.
3. *Final value* of the control parameter: it is fixed at some small value, generally related to the smallest possible difference in cost between two neighboring solutions.
4. *Length* L_k of Markov chain: it is fixed by some number that may be related to the size of the neighborhoods in the problem instance at hand.

Some year later, the optimality of a logarithmic cooling schedule has been proved (Hajek, 1988) and efficient algorithms were developed for computing the initial temperature T_o.

In the following we illustrate a summary of the algorithm proposed by Ben-Ameur (Ben-Ameur, 2004), the author also uses in the optimization algorithm HGO, to compute the optimal initial temperature. It requires the definition of an initial acceptance probability χ_0, i.e. the percentage of pejorative accepted transitions at the beginning of the process. A commonly used value is $\chi_0 = 0.8$. Let t be a strictly positive transition and let $\max_t$ (resp. $\min_t$) be the state after (resp. before) the transition. As we assumed that the transition is strictly positive, then $E_{\max_t} > E_{\min_t}$. Using the generation strategy (8.26), the acceptance probability is given by:

$$\chi(T) = \frac{\sum_{t \text{ positive}} \pi_{\min_t} |\Gamma(\min_t)|^{-1} \exp\left[-\left(E_{\max_t} - E_{\min_t}\right)/T\right]}{\sum_{t \text{ positive}} \pi_{\min_t} |\Gamma(\min_t)|^{-1}} \tag{8.29}$$

Note that $\pi_{\min_t} |\Gamma(\min_t)|^{-1}$ represents the probability to generate a transition t when the energy states are distributed in conformance with the stationary distribution (8.28). Moreover, $\exp\left[-\left(E_{\max_t} - E_{\min_t}\right)/T\right]$ is the probability to accept a positive transition t. Thus, $\chi(T)$ is the conditional expectation of the acceptance of positive transitions. We will use an estimation $\hat{\chi}(T)$ of this acceptance probability, based on a random set S of positive transitions. $\hat{\chi}(T)$ is defined as follows:

$$\hat{\chi}(T) = \frac{\sum_{t \in S} \pi_{\min_t} |\Gamma(\min_t)|^{-1} \exp\left[-\left(E_{\max_t} - E_{\min_t}\right)/T\right]}{\sum_{t \in S} \pi_{\min_t} |\Gamma(\min_t)|^{-1}} = \frac{\sum_{t \in S} \exp\left(-E_{\max_t}/T\right)}{\sum_{t \in S} \exp\left(-E_{\min_t}/T\right)} \tag{8.30}$$

We are looking for a temperature T_0 such that $\chi(T_0) = \chi_0$, where $\chi_0 \in]0, 1]$ is the desired acceptance probability. The simple iterative method proposed by (Ben-Ameur, 2004) consist in consider $\hat{\chi}(T)$ instead of $\chi(T)$. First, we randomly generate a set of positive transitions S, generating some states and a neighbor for each state. The energies $E_{\max_t}$ and $E_{\min_t}$ corresponding with the states of the subset S are stored. Then we use the Johnson formula (Johnson, Aragon, McGeoch, & Schevon, 1989) for computing an initial temperature T_1:

$$T_1 = -\frac{\sum_{t \in S}\left(E_{\max_t} - E_{\min_t}\right)}{S \ln(\chi_0)} \tag{8.31}$$

T_1 may be far from T_0; to find T_0 we use the recursive formula:

$$T_{n+1} = T_n \left[\frac{\ln\left(\hat{\chi}(T_n)\right)}{\ln(\chi_0)}\right]^{\frac{1}{p}} \tag{8.32}$$

where p is a real number ≥ 1. When $\hat{\chi}(T_n)$ becomes close to χ_0 we can stop: T_n is a good approximation of the desired temperature T_0. Fig. 8.5 shows the flow chart of the algorithm to find the optimal initial temperature. $\varepsilon_1$and ε_2 are the relative errors admitted on the values of χ and T_0, while ΔS is the increment of the number of positive transitions considered until T_0 becomes invariant.

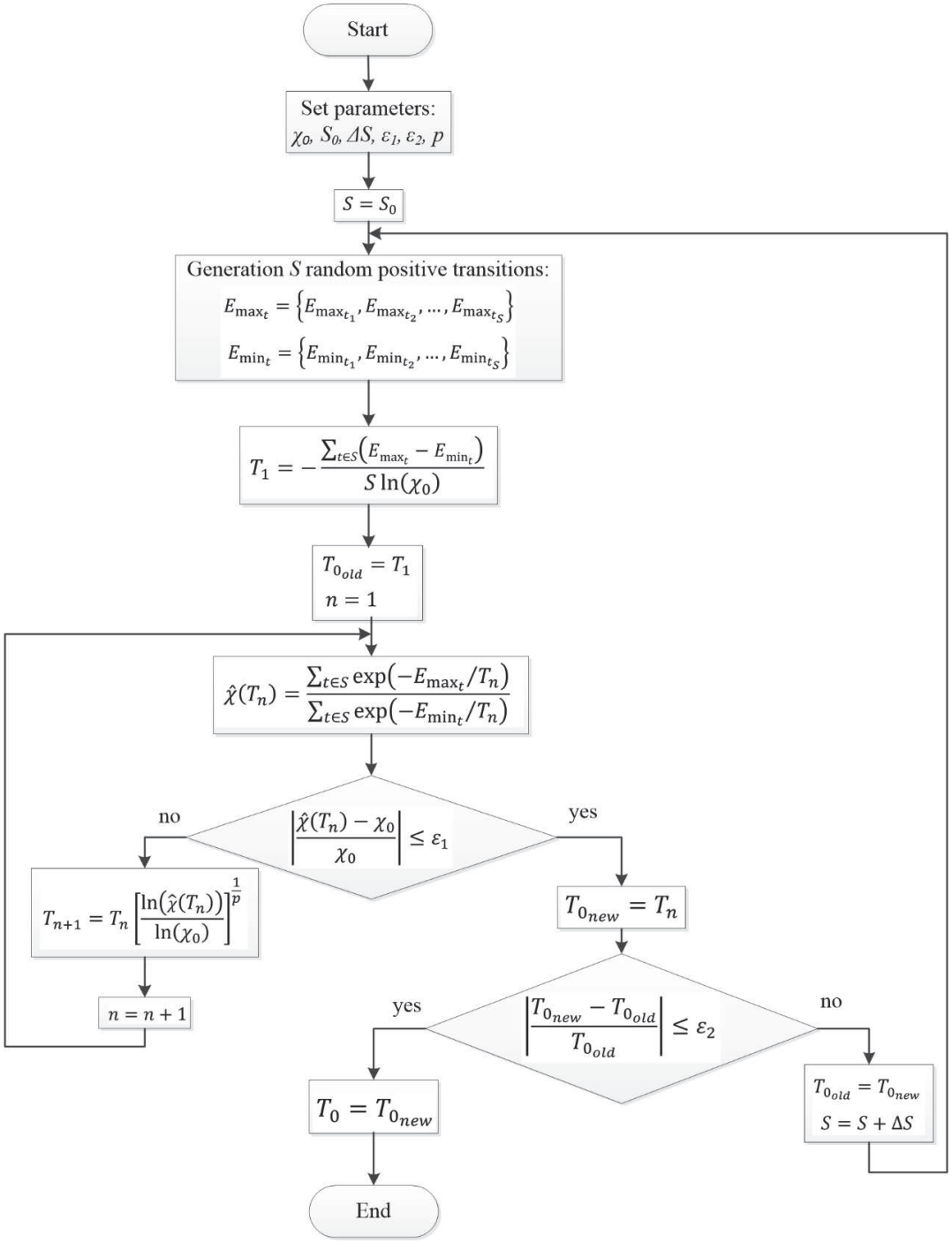


Fig. 8.5, The Optimal Initial Temperature flow chart.

8.6 Notes of Information Theory

In this paragraph, the author introduces some basic notions of information theory, useful for understanding the meaning of the concept of mutual information, introduced in chapter 4. The basic concept to introduce is the one of *self-information*, I, a measure of the information content associated with a single value of a discrete random variable.

Let us consider the simple case of a random binary variable, e.g. the toll of a coin which may take only two values: head, that will be identified by a 0, and cross, identified by 1. To identify the state of the system, i.e. to know the whole information about the state of the coin, we need to fill a box with the 0 or the 1. We just need to fill one box, i.e. we need one bit (bit because the variable is binary) of information. Now assume that we toss the same coins two times. In this case four (2^2) possible outcomes can be observed (0, 0) or (0, 1) or (1, 0) or (1, 1). In this case, we need to fill-in two different boxes i.e. we need 2 bits. If we toss the same coin N times we will need to fit N boxes i.e. we will need to know N bits of information. So, we note that the information is *additive*, although the number of states increase as 2^N. Moreover, we require that the information is *positive*. Note that

$$I = \log_2 2^N = N \tag{8.33}$$

and, observe that the probability of each given state of the system for N coins is

$$p = \prod_{k=1}^{N} \bar{p} = \bar{p}^N \tag{8.34}$$

where $\bar{p} = 1/2$ is the probability to get head or cross in a single toll of a single coin. If we take the *log* of the above relation, we have

$$\log_2 p = \log_2 \bar{p}^N = N \log_2 \bar{p} = -N \tag{8.35}$$

$$I = -\log_2 p = -\log_2 \prod_{k=1}^{N} \bar{p} = \sum_{k=1}^{N} -\log_2 \bar{p} = N \tag{8.36}$$

The mean value of provided information per bit is then:

$$H = \frac{I}{N} = \frac{1}{N} \sum_{k=1}^{N} -\log_2 \bar{p} \tag{8.37}$$

We also rearranging the terms can rephrase I as

$$I = -N \sum_{k=1}^{2} \bar{p} \log_2 \bar{p} \tag{8.38}$$

i.e.

$$H = -\sum_{k=1}^{2} \bar{p} \log_2 \bar{p} \tag{8.39}$$

Consider that the quantity I can be interpreted as the uncertainty of the state of the N string, and in fact we need to measure (or specify) N bit values to complete specify the state of the system of N trials. Also, note that when the coins are the same, then N successive tolls of the same coins are equivalent to the tolls of N coins. If the coins are fair, we have $p_2 = 1 - p_1 = \bar{p}$ (where p_1 is the probability of head i.e. of 0 and p_2 the probability of cross i.e. 1), and we may think of generalizing the above equation as (we assume very large $N \gg 1$)

$$I = -N \sum_{k=1}^{2} p_k \log_2 p_k = -\sum_{k=1}^{2} n_k \log_2 p_k \tag{8.40}$$

where $n_1 = p_1 N$ is the number of times you find heads and $n_2 = p_2 N$ is the number of times you find cross. Rearranging the terms, we get

$$I = -\sum_{k=1}^{2} n_k \log_2 n_k + \sum_{k=1}^{2} n_k \log_2 N \tag{8.41}$$

$$I = -\sum_{k=1}^{2} n_k \log_2 n_k + N \log_2 N \tag{8.42}$$

from which

$$H = \frac{I}{N} = -\sum_{k=1}^{2} p_k \log_2 p_k = -\sum_{k=1}^{2} p_k \log_2 n_k + \log_2 N \tag{8.43}$$

We will show now that the above definition $I = -N \sum_{k=1}^{2} p_k \log_2 p_k$ holds true also for general values of p_k. But first observe that this definition satisfies an interesting property (besides the fact that it is positive and extensive): if $p_1 \to 0$, i.e. $p_2 \to 1$ we get $I = 0$.
This is what we want to happen as if we know that the probability of head is 0 there is no information gain (no surprise) when we experiment during the toll that we get cross: no information is provided during the toll. Also, it can be easily shown that the amount of information we get during the experiment takes its maximum value just when the coin is fair, i.e. when $p_1 = p_2 = \bar{p} = 1/2$, this is also what we want as we are most surprised by the outcome of the toll when the coin is fair and there is no a priory preference for head or cross.

In order to understand why we can generalize the definition of I, eq. (8.40), to the case when $p_k \neq 1/2$ consider the N tolls of the same unfair coin and that we know a-priory which are the probabilities p_k (this is a very important point as if we do not know that the coins are

fair then our a priori probability would be $1/2$ and the gain of information after the toll would be the same as for fair coins), or equivalently the contemporary toll of N equally coins. We assume that N is very large.

Let us calculate what is the gain of information that we get after the toll, which is the same as the amount of uncertainty before the coins are tolled. Since N is very large, because of the central limit, we may assume the number of heads is $n_1 = Np_1$ and therefore the number of crosses is $n_2 = Np_2$. Therefore, this time we are considering strings of n_1 heads and n_2 cross where the order of the heads and cross is unknown a priori and of course can be considered equally distributed. Thus, specify the state of the systems means that we have to specify which one out of the total number $N_T = 2^M$ of possible ordering of n_1 heads and n_2 cross we get when tolling the coins, i.e. we can say that the total information gain after the measurement (or the amount of uncertainty before the measurement) is

$$I = \log_2 N_T = M \tag{8.44}$$

Observe that in general M can be real and not integer. So, the point is to find the total number of possible configurations; this is simply given by recalling that N_T is not just given by the number of permutations of a string of N figures which is $N!$ as interchanging the order of those figures which have taken the same outcome (head of cross) will not modify the order of the resulting string of toll outcomes. Therefore $N_T < N!$ and can be obtained by dividing it for the number of permutations $n_1!$ of n_1 heads and dividing once more by the $n_2!$ permutation of n_2 heads.

$$N_T = \frac{N!}{n_1!\, n_2!} \tag{8.45}$$

Therefore, we get

$$I = \log_2 \frac{N!}{n_1!\, n_2!} = \log_2 N! - \log_2 n_1! - \log_2 n_2! \tag{8.46}$$

Now we need to calculate the logarithm of the factorial. To this end we use the Stirling approximation for $n \gg 1$, i.e.

$$\log_a n! = n \log_a n - n \log_a e + \frac{1}{2}\log_a n + \frac{1}{2}\log_a(2\pi) \tag{8.47}$$

replacing the above formula into the preceding one we get

$$I = N \log_a N - N \log_a e + \frac{1}{2}\log_a N + \frac{1}{2}\log_a(2\pi) + \tag{8.48}$$

$$-n_1 \log_a n_1 + n_1 \log_a e - \frac{1}{2}\log_a n_1 - \frac{1}{2}\log_a(2\pi) +$$

$$-n_2 \log_a n_2 + n_2 \log_a e - \frac{1}{2}\log_a n_2 - \frac{1}{2}\log_a(2\pi).$$

recalling that $n_1 + n_2 = N$ and that $n_1 = p_1 N$ and $n_2 = p_2 N$ we also get

$$I = N \log_a N - Np_1 \log_a n_1 - Np_2 \log_a n_2 + \frac{1}{2}\log_a N - \frac{1}{2}\log_a n_2 - \frac{1}{2}\log_a n_1 - \frac{1}{2}\log_a(2\pi)$$

$$I = Np_1 \log_a N + Np_2 \log_a N - Np_1 \log_a n_1 - Np_2 \log_a n_2 + \frac{1}{2}\log_a N - \frac{1}{2}\log_a n_2 - \frac{1}{2}\log_a n_1 - \frac{1}{2}\log_a(2\pi)$$

$$I = -Np_1 \log_a p_1 - Np_2 \log_a p_2 - \frac{1}{2}\log_a p_2 - \frac{1}{2}\log_a p_1 - \frac{1}{2}\log_a N - \frac{1}{2}\log_a(2\pi)$$

So, the average information per coin is

$$H = \frac{I}{N} = -p_1 \log_a p_1 - p_2 \log_a p_2 - \frac{1}{2N}\log_a p_2 - \frac{1}{2N}\log_a p_1 - \frac{1}{2N}\log_a N - \frac{1}{2N}\log_a(2\pi) \quad (8.49)$$

and recalling that $a = 2$ in our case and taking the limit for $N \to \infty$ we get

$$H = -\sum_{k=1}^{2} p_k \log_2 p_k \quad (8.50)$$

which proves our statement. Of course, more generally, if the number of different states the single variable can take is larger than 2, say s different states, then

$$N_T = \frac{N!}{n_1!\, n_2! \ldots n_s!} \quad (8.51)$$

and the above formula becomes

$$H = -\sum_{k=1}^{s} p_k \log_2 p_k \quad (8.52)$$

which is the information associated to one single toll of the given stochastic variable, we can also call it the entropy per single toll of the given variable. Observe that H is the average information associated to a single toll of the given discrete stochastic variable.

Now consider to discrete variables $X = \{x_k\}$ and $X = \{y_h\}$ and suppose we know the joint probability distribution $p_{kh} = p(x_k, y_h)$. We also know that

$$p_{kh} = p(x_k, y_h) = p(x_k|y_h)p(y_h) = p(y_h|x_k)p(x_k) \quad (8.53)$$

and

$$p(x_k) = \sum_h p(x_k, y_h) = \sum_h p(x_k|y_h)p(y_h) \quad (8.54)$$

Now observe that given the value of y_h if we do a certain number of tolls of the variable X we will found the probability distribution $p(x_k|y_h)$ therefore the average information associated to X is

$$H(X|y_h) = -\sum_k p(x_k|y_h) \log_2 p(x_k|y_h) \quad (8.55)$$

Now we want to calculate the average information we get when we know the distribution of y_h values. We get

$$H(X|Y) = \langle H(X|y_h)\rangle = -\sum_{kh} p(x_k|y_h)\, p(y_h) \log_2 p(x_k|y_h) = -\sum_{kh} p(x_k, y_h)\ \log_2 \frac{p(x_k, y_h)}{p(y_h)} \quad (8.56)$$

which, can be interpreted as the amount of uncertainty that you have on the variable X when you know the variable Y. Observe that this differs from

$$H(X) = -\sum_{k} p(x_k)\ \log_2 p(x_k) = -\sum_{kh} p(x_k, y_h) \log_2 p(x_k) \quad (8.57)$$

we expect $(X) > H(X|Y)$, so, if we take the difference we have:

$$MI(X,Y) = H(X) - H(X|Y)$$

$$MI(X,Y) = -\sum_{kh} p(x_k, y_h)\ log_2\, p(x_k) + \sum_{kh} p(x_k, y_h)\ log_2 \frac{p(x_k, y_h)}{p(y_h)}$$

$$MI(X,Y) = \sum_{kh} p(x_k, y_h)\ log_2 \frac{p(x_k, y_h)}{p(x_k)p(y_h)} \quad (8.58)$$

This is what we call the *mutual information* i.e. the difference between the uncertainty of the variable X when you do not know the variable Y and the uncertainty on the variable X when you know Y, this means that you have reduced the uncertainty on variable X if you know the variable Y, i.e. you gained some info about X. Observe that if the two variables are independent the mutual information is zero.

$$MI(X,Y) = \sum_{kh} p(x_k, y_h)\ log_2\, p(x_k, y_h) - \sum_{kh} p(x_k, y_h)\ log_2\, p(x_k) - \sum_{kh} p(x_k, y_h)\ log_2\, p(y_h)$$

$$MI(X,Y) = H(X) + H(Y) - H(X,Y) \quad (8.59)$$

8.7 The rescaled NK Landscape

According to the *NK* model (Kauffman & Levin, 1987), the payoff $V(\boldsymbol{d})$ associated with a decision vector $\boldsymbol{d}$, is calculated as the average of statistically independent contributions W_k, uniformly drawn in the interval $[0,1]$.

$$V(\boldsymbol{d}) = N^{-1} \sum_k W_k \tag{8.60}$$

For large N the distribution of $V(\boldsymbol{d})$ converge in probability to a Gaussian distribution

$$p(V) = \frac{1}{\sigma_V \sqrt{2\pi}} \exp\left[-\frac{1}{2}\frac{(V-\bar{V})^2}{\sigma_V^2}\right] \tag{8.61}$$

with mean $\bar{V} = \bar{W}$ (where $\bar{W} = 0.5$ is the expected value of each contribution W_k), and variance

$$\sigma_V^2 = \langle (V-\bar{V})^2 \rangle = \frac{1}{N^2} \sum_{j,k} \langle (W_j - \bar{W})(W_k - \bar{W}) \rangle = \frac{1}{N^2} \sum_{j,k} \delta_{jk} \langle (W_j - \bar{W})^2 \rangle$$

$$\sigma_V^2 = \frac{1}{N^2} \sum_j \langle (W_j - \bar{W})^2 \rangle = \frac{N\sigma_W^2}{N^2} = N^{-1}\sigma_W^2 \tag{8.62}$$

where σ_W^2 is the variance of each stochastic contribution W_k.

Note that this property holds true for all K values as the correlation between the decisions d_j does not affect individual fitness values.

Observing eq. (8.62), it is worth noting that the fitness variance σ_V^2 results to be inversely proportional to the size of the problem N. In order to relate the performance of the HGO algorithm between *NK* landscapes of different sizes N, we need to make them comparable, rescaling the fitness landscape, i.e. changing the variable V to ζ, such that the new random variable ζ is always Gaussian distributed around the average value $\bar{V}$, but with a variance $\sigma_\zeta^2 = \sigma_W^2$, independent by N.

$$p(\zeta) = \frac{1}{\sigma_\zeta \sqrt{2\pi}} \exp\left[-\frac{1}{2}\frac{(\zeta-\bar{V})^2}{\sigma_\zeta^2}\right] \tag{8.63}$$

To find the relation between ζ and V, observe that

$$p(V)dV = p(\zeta)d\zeta \tag{8.64}$$

from which we have

$$\frac{d\zeta}{dV} = \frac{p(V)}{p(\zeta)} = \frac{\sigma_\zeta}{\sigma_V} \exp\left\{-\frac{1}{2}\left[\frac{(V-\bar{V})^2}{\sigma_V^2} - \frac{(\zeta-\bar{V})^2}{\sigma_\zeta^2}\right]\right\} \tag{8.65}$$

As the variables V and ζ, are both Gaussian distributed around the same average value $\bar{V}$, there must be a linear relation between $V - \bar{V}$ and $\zeta - \bar{V}$.

$$\zeta - \bar{V} = a(V - \bar{V}) \tag{8.66}$$

By replacing eq. (8.69) in eq. (8.68), we get:

$$a = \frac{\sigma_\zeta}{\sigma_V} \exp\left\{-\frac{1}{2}\left[\frac{(V-\bar{V})^2}{\sigma_V^2} - \frac{a^2(V-\bar{V})^2}{\sigma_\zeta^2}\right]\right\}$$

$$a = \frac{\sigma_\zeta}{\sigma_V} \exp\left\{-\frac{1}{2}\frac{(V-\bar{V})^2}{\sigma_V^2}\left[1 - \frac{a^2\sigma_V^2}{\sigma_\zeta^2}\right]\right\} \tag{8.67}$$

To be a constant, the right-hand side of the above equation cannot be dependent on V, so

$$1 - \frac{a^2\sigma_V^2}{\sigma_\zeta^2} = 0 \Longrightarrow a = \frac{\sigma_\zeta}{\sigma_V} \tag{8.68}$$

Now, remembering that $\sigma_\zeta = \sigma_\mathrm{W}$ and $\sigma_V = \sigma_W/\sqrt{N}$, the constant takes value $a = \sqrt{N}$.

In conclusion, we get the following rescaling relation:

$$\zeta = \bar{V} + \sqrt{N}(V - \bar{V}) \tag{8.69}$$

9 References

Aarts, E. E. (1985). Statistical cooling: A general approach to combinatorial optimization problems. *Philips Journal of research, 40*(4), 193.

Aarts, E. H., & Korst, J. H. (1989). *Simulated Annealing and Boltzmann Machines.* Chichester: Wiley.

Aarts, E., Korst, J., & Michiels, W. (2014). Simulated annealing. In *Search methodologies* (p. 265-285). Springer US.

Arganda, S., Pérez-Escudero, A., & de Polavieja, G. G. (2012). A common rule for decision making in animal collectives across species. *Proceedings of the National Academy of Sciences of the United States of America*, *109*(50), 20508–20513. https://doi.org/10.1073/pnas.1210664109

Arostegui, M. A., Kadipasaoglu, S. N., & Khumawala, B. M. (2006). An empirical comparison of tabu search, simulated annealing, and genetic algorithms for facilities location problems. *International Journal of Production Economics, 103*, 742-754.

Arumugam, Rao, & Tan. (2009:). novel and effective particle swarm optimization like algorithm with extrapolation technique. *Applied Soft Computing*, 308–320.

Atyabi, A., & Powers, D. (2013). Cooperative area extension of PSO: Transfer learning vs. uncertainty in a simulated swarm robotics. *Informatics in Control Automation and Robotics*, (p. 177–184).

Aytug, H., Khouja, M., & Vergara, F. E. (2003). Use of genetic algorithms to solve production and operations management problems: a review. *International Journal of Production Research, 41*, 3955-4009.

Bai, Q. (2010). Analysis of Particle Swarm Optimization Algorithm. *Computer and Information Science*, 180–184.

Baldassarre, G., Parisi, D., & Nolfi, S. (2006). Distributed coordination of simulated robots based on self-organization. *Artificial Life, 12*(3), 289-311.

Balkundi, P., and Harrison, D. A., Ties, leaders, and time in teams: Strong inference about network structure's effects on team viability and performance, *Academy of Management Journal* **49**(1) (2006) 49-68.

Ball, P. (1999). Shark skin and other solutions. *Nature, 400*, 507–508.

Bao, L., & Zeng, J. (2009). Comparison and Analysis of the Selection Mechanism in the Artificial Bee Colony. *Algorithm. International Conference on Hybrid Intelligent System*, p. 411–416.

Barnett, L. L. (2013). Information flow in a kinetic Ising model peaks in the disordered phase. *Physical review letters, 111*(17), 177203.

Beckers, R., Deneubourg, J. L., Goss, S., & Pasteels, J. M. (1990). Collective decision making through foodr ecruitment. *Insectes Sociaux, 37*, 258–267.

Ben-Ameur, W. (2004). Computing the initial temperature of simulated annealing. *Computational Optimization and Applications, 29*(3), 369-385.

Beni, G. a. (1989). Swarm Intelligence in Cellular Robotic Systems. *Proceed. NATO Advanced Workshop on Robots and Biological Systems*, (p. 26-30). Tuscany, Italy.

Ben-Jacob, E., Cohen, I., & Levine, H. (2000). Cooperative self-organization of microorganisms. *Advances in Physics, 49*, 395–554.

Beshers, S. N., & Fewell, J. H. (2001). Models of division of labor in social insects. *Annual Review of Entomology,, 46*(1), 413–440.

Billinger, S., Stieglitz, N., & Schumacher, T. R. (2014). Search on Rugged Landscapes: An Experimental Study. *Organization Science*, *25*(1), 93–108. https://doi.org/10.1287/orsc.2013.0829

Binder, K. (1978). *Monte Carlo Methods in Statistical Physics.* Berlin: Springer.

Binitha, S., & Sathya, S. S. (2012). A survey of bio inspired optimization algorithms. *International Journal of Soft Computing and Engineering, 2*, 137-151.

Boccaletti, S., Bianconi, G., Criado, R., del Genio, C. I., G??mez-Garde??es, J., Romance, M., … Zanin, M. (2014). The structure and dynamics of multilayer networks. *Physics Reports*. https://doi.org/10.1016/j.physrep.2014.07.001

Bolaji, A., Khader, A., Al-Betar, M., & Awadallah, M. (2013). Artificial bee colony algorithm, its variants and applications: A survey. *Journal of Theoretical Applied Information Technology, 47*, 434–459.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (2000). Inspiration for optimization from social insect behaviour. *Nature*, *406*(6791), 39–42. https://doi.org/10.1038/35017500

Bonabeau, E., Theraulaz, G., & Deneubourg, J. L. (1998). Fixed response threshold and the regulation of division of labor in insect societies. *Bulletin of Mathematical Biology, 60*, 753–807.

Bonabeau, E., Theraulaz, G., Deneubourg, J. L., Aron, S., & Camazine, S. (1997). Self-organization in social insects. *Trends in Ecology and Evolution, 12*(5), 188–193.

Bordogna, C. M., & Albano, E. V. (2007). Dynamic behavior of a social model for opinion formation. *Physical Review E*, 76(6), 061125.

Borgatti, S. P., and Foster, P. C., The network paradigm in organizational research: A review and typology, *Journal of management* **29**(6) (2003) 991-1013.

Brambilla, M. F. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence, 7*(1), 1-41.

Bratton, D., & Kennedy, J. (2007). Defining a Standard for Particle Swarm Optimization. *IEEE Swarm Intelligence Symposium*, (p. 120–127).

Brummitt, C. D. (2015). Jigsaw percolation: What social networks can collaboratively solve a puzzle? *The Annals of Applied Probability, 25*(4), 2013-2038.

Brush, S. G. (1967). History of the Lenz-Ising model. *Reviews of Modern Physics*, *39*(4), 883–893. https://doi.org/10.1103/RevModPhys.39.883

Buhl, J., Deneubourg, J. L., Grimal, A., & Theraulaz, G. (2005). Self-organized digging activity in ant. *Behavioral Ecology and Sociobiology, 58*(1), 9–17.

Buhl, J., Sumpter, D. J., Couzin, I. D., Hale, J. J., Despland, E., Miller, E. R., & Simpson, S. J. (2006). From disorder to order in marching locusts. *Science,, 312*(5778), 1402–1406.

Burd, M. (2006). Ecological consequences of traffic organisation in ant societies. *Physica A: Statistical and Theoretical Physics, 372*(1), 124–131.

Camazine, S., Deneubourg, J. L., Franks, N. R., Sneyd, J., Theraulaz, G., & Bonabeau, E. (2001). *Selforganization in biological systems.* Princeton: Princeton University Press.

Campo, A., Nouyan, S., Birattari, M., Groß, R., & Dorigo, M. (2006). Enhancing cooperative transport using negotiation of goal direction. *BNAIC 2006.* University of Namur.

Carbone, G., & Giannoccaro, I. (2015). Model of human collective decision-making in complex environments. *The European Physical Journal B, 88*(12), 339-348.

Castellano, C., Fortunato, S., & Loreto, V. (2009). Statistical physics of social dynamics. *Reviews of Modern Physics*, *81*(2), 1–58. https://doi.org/10.1103/RevModPhys.81.591

Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *Journal of optimization theory and applications*, 41-51.

Chaudhry, S. S., & Luo, W. (2005). Application of genetic algorithms in production and operations management: a review. *International Journal of Production Research, 43*, 4083-4101.

Chiao Mei, F., Phon-Amnuaisuk, S., Alias, M., L. P., & Adaptive, G. (2008). An essential ingredient in high-level synthesis. *IEEE Congress on Evolutionary Computation*, (p. 3837–3844).

Chung, Y., and Jackson, S. E., The Internal and External Networks of Knowledge-Intensive Teams the Role of Task Routineness, *Journal of Management* **39**(2) (2013) 442-468.

Clark, K. B., and Wheelwright, S. C., vol. **34**(3) (1992) 9–28.

Clément, R. J. (2013). Collective cognition in humans: groups outperform their best members in a sentence reconstruction task. *PLoS One, 8*(10), e77943.

Clerc, M., & Kennedy, J. (2002). The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space. *IEEE Transactions on Evolutionary Computation, 6*(1), 58–73.

Conradt, L. &. (2003). Group decision-making in animals. *Nature, 421*(6919), 155-158.

Conradt, L. (2012). Models in animal collective decision-making: information uncertainty and conflicting preferences. *Interface Focus*, *2*(2), 226–40. https://doi.org/10.1098/rsfs.2011.0090

Couzin, I. D. (2005). Effective leadership and decision-making in animal groups on the move. *Nature, 433*(7025), 513-516.

Couzin, I. D. (2009). Collective cognition in animal groups. *Trends in Cognitive Sciences*. https://doi.org/10.1016/j.tics.2008.10.002

Couzin, I. D., & Franks, N. R. (2003). Self-organized lane formation and optimized traffic flow in army ants. *Proceedings of the Royal Society B Biological Sciences, 270*, 139–146.

Dawkins, R. (1996). *Climbing Mount Improbable.*

De Domenico, M., Solé-Ribalta, A., Cozzo, E., Kivelä, M., Moreno, Y., Porter, M. A., … Arenas, A. (2013). Mathematical Formulation of Multilayer Networks. *Physical Review X*, *3*(4), 41022. https://doi.org/10.1103/PhysRevX.3.041022

De Jong, K., & Spears, W. (1992). A formal analysis of the role of multi-point crossover in genetic algorithms. *Annals of Mathematics and Artificial Intelligence, 5*, 1–26.

DelValle, Venayagamoorthy, Mohagheghi, Hernandez, & Harley. (2008). Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power System. *IEEE Trans Evolutionary Computer*, 171–195.

Deneubourg, J. L., Goss, S., Pasteels, J. M., Fresneau, D., & Lachaud, J. P. (1987). Self-organization mechanisms in ant societies (II): learning in foraging and division of labor. *Proceedings of the from individual to collective behavior in social insects conference*, (p. 177–196). Birkhäuser.

Detrain, C., & Deneubourg, J. L. (2006). Self-organized structures in a superorganism: do ants "behave" like molecules? *Physics of Life Reviews, 3*(3), 162–187.

DiMaggio, P. J., & Anheier, H. K. (1990). The Sociology of Nonprofit Organizations. *Annual Review of Sociology*, *16*(1990), 137–159. https://doi.org/10.1146/annurev.so.16.080190.001033

Dimaggio, P. J., & Powell, W. W. (1983). The iron cage revisited: Institutional isomorphism and collective rationality in organizational fields. *American Sociological Review*, 48(2), 147-160.

Doolen, T. L., Hacker, M. E., and Van Aken, E. M., The impact of organizational context on work team effectiveness: A study of production team, *Engineering Management, IEEE Transactions on* **50**(3) (2003) 285-296.

Dorigo, & Gambardella. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *Transaction on Evolutionary Computation, 1*, 53–66.

Dorigo, Birattari, & Stutzle. (2006). Ant Colony Optimization. *Computational Intelligence Magazine*, 28–39.

Dorigo, M. (1992). *Optimization, learning and natural algorithms.* Politecnico di Milano, Milan: Ph.D. Thesis.

Dorigo, M., & Gambardella, L. M. (1997). Ant colonies for the travelling salesman problem. *Biosystems*, *43*(2), 73–81. https://doi.org/Doi 10.1016/S0303-2647(97)01708-5

Dorigo, M., Di Caro, G., & Gambardella, L. (1999). Ant algorithms for discrete optimization. *Artificial life, 5*, 137-172.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *26*(1), 29–41. https://doi.org/10.1109/3477.484436

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, 26*, 29-41.

Dorogovtsev, S. N., Goltsev, A. V., & Mendes, J. F. (2002). Ising model on networks with an arbitrary distribution of connections. *Phys. Rev. E, 66*, 016104.

Dussutour, A., Deneubourg, J. L., & Fourcassié, V. (2005). Amplification of individual preferences in a social. *Proceedings of the Royal Society B Biological Sciences, 272*, 705–714.

Easley, D., & Kleinberg, J. (2010). Networks , Crowds , and Markets : Reasoning about a Highly Connected World. *Science*, *81*, 744. https://doi.org/10.1017/CBO9780511761942

Eberhart, R., & Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. *Congress on Evolutionary Computation*, (p. 84–88).

Engel, D., Woolley, A. W., Jing, L. X., Chabris, C. F., & Malone, T. W. (2014). Reading the mind in the eyes or reading between the lines? Theory of mind predicts collective intelligence equally well online and face-to-face. *PloS one, 9*, e115212.

Farhi, E., Goldstone, J., Gutmann, S., Lapan, J., Lundgren, A., & Preda, D. (2001). A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science, 292*, 472-475.

Faria, J. J. (2009). Navigation in human crowds; testing the many-wrongs principle. *Animal Behaviour, 78*(3), 587-591.

Feller, W. (1950). *An Introduction to Probability Theory and Its Applications 1.* New York: Wiley.

Ferretti, I., Zanoni, S., & Zavanella, L. (2006). Production--inventory scheduling using Ant System metaheuristic. *International Journal of Production Economics, 104*, 317-326.

Franks, N. R. (1986). Teams in social insects: group retrieval of prey by army ants (Eciton burchelli, Hymenoptera: Formicidae). *Behavioral Ecology and Sociobiology, 18,*, 425–429.

Galam, S. &. (2015). Two-dimensional Ising transition through a technique from two-state opinion-dynamics models. *Physical Review E, 91*(1), 012108.

Galam, S. (1997). Rational group decision making: A random field Ising model at T= 0. *Physica A: Statistical Mechanics and its Applications, 238*(1), 66-80.

Garey, M., & Johnson, D. S. (1979). *Computers and Intractability: a guide to the theory of NP-completeness.* San Francisco: W. H. Freeman and Company.

Garnier, S., Gautrais, J., & Theraulaz, G. (2007). The biological principles of swarm intelligence. *Swarm Intelligence*, *1*(1), 3–31. https://doi.org/10.1007/s11721-007-0004-y

Gautrais, J., Michelena, P., Sibbald, A., Bon, R., & Deneubourg, J.-L. (2007). Allelomimetic synchronisation in Merino sheep. *Animal Behaviour*.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, 721-741.

Getling, A. V. (1998). Structures and Dynamics. In *Advanced series in nonlinear dynamics* (Vol. 11).

Gillespie, D. T. (1976). A general method for numerically simulation the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, *22*(4), 403–434. https://doi.org/10.1016/0021-9991(76)90041-3

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry, 81*(25), 2340-2361.

Glauber, R. J. (1963). Time-dependent statistics of the Ising model. *Journal of mathematical physics, 4*(2), 294-307.

Goldberg, D. (1987). *Genetic Algorithms in Optimization, Search and Machine Learning.* Addison Wesley.

Golden, B. L., & Skiscim, C. C. (1986). Using simulated annealing to solve routing and location problems. *Naval Research Logistics Quarterly, 33*, 261-279.

Gordon, D. M. (1996). The organization of work in social insect colonies. *Nature, 380*(6570), 121–124.

Goss, S., Aron, S., Deneubourg, J. L., & Pasteels, J. M. (1989). Self-organized shortcuts in the Argentine ant. *Naturwissenschaften, 76*, 579–581.

Grassé, P. P. (1959). La reconstruction du nid et les coordinations inter-individuelles chez Bellicositermes Natalensis et Cubitermes sp. La théorie de la stigmergie : essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux, 6*, 41–81.

Grünbaum, D. (1998). Schooling as a strategy for taxis in a noisy environment. *Evolutionary Ecology, 12*(5), 503-522.

Grünbaum, D., Viscido, S. V., & Parrish, J. K. (2005). Extracting interactive control algorithms from group dynamics of schooling fish. *Proceedings of the cooperative control conference* (p. 103–117). Heidelberg: Springer.

Hajek, B. (1988). Cooling schedules for optimal annealing. *Mathematics of operations research, 13*, 311–329.

He, L., Tong, X., & Huang, S. (2012). A Glowworm Swarm Optimization Algorithm with Improved Movement Rule. *Fifth International Conference on Intelligent Networks and Intelligent Systems*, (p. 109–112).

Helbing, D., Molnàr, P., Farkas, I. J., & Bolay, K. (2001). Self-organizing pedestrian movement. *Environment and Planning B: Planning and Design, 28*(3), 361–383.

Higgins, F., Tomlinson, A., & Martin, K. M. (2009.). Survey on security challenges for swarm robotics. *Proceedings of the 5th International Conference on Autonomic and Autonomous Systems (ICAS '09)* (p. 307–312). Los Alamitos, CA, USA: IEEE Computer Society.

Hillis, D. (1990). Co-evolving parasites improve simulated evolution in an optimization procedure. *Physica D, 42*, 228–234.

Holland, J. (1992). *Genetic Algorithms.* Scientific American Journal.

Hölldobler, B., & Wilson, E. O. (1990). *The ants.* Cambridge: Harvard University Press.

Horwitz, S. K., and Horwitz, I. B., The effects of team diversity on team outcomes: A meta-analytic review of team demography, *Journal of management* **33**(6) (2007) 987-1015.

Iocchi, L., Nardi, D., & Salerno, M. (2001). Reactivity and deliberation: a survey on multi-robot systems. *Balancing Reactivity and Social Deliberation in Multi-Agent Systems. From RoboCup to Real-World Applications* (p. 9–32). Berlin, Germany: Springer.

Isaacson, D., & Madsen, R. (1976). *Markov Chains.* New York: Wiley.

Ising, E. (1925). Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei, 31*(1), 253-258.

Jackson, S. E., DeNisi, A., and Hitt, M. A., *Managing knowledge for sustained competitive advantage: Designing strategies for effective human resource management* **21** (John Wiley and Sons, 2003).

Jackson, S. E., Joshi, A., & Erhardt, N. L. (2003). Recent research on team and organizational diversity: SWOT analysis and implications. *Journal of management, 29*(6), 801-830.

Janson, S., Middendorf, M., & Beekman, M. (2005). Honeybee swarms: how do scouts guide a swarm of uninformed bees? *Animal Behaviour, 70*(2), 349–358.

Jayaraman, V., & Ross, A. (2003). A simulated annealing methodology to distribution network design and management. *European Journal of Operational Research, 144*, 629-645.

Johnson, D., Aragon, C., McGeoch, L., & Schevon, C. (1989). Optimization by simulated annealing: An experimental evaluation; part I, graph partitioning. *Operations Research, 37*, 865–892.

Jones, C., & Matarić, M. (2003). Adaptive division of labor in large-scale minimalist multi-robot systems. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, (p. 1969–1974). Las Vegas.

Kadowaki, T., & Nishimori, H. (1998). Quantum annealing in the transverse Ising model. *Physical Review E, 58*, 5355.

Karaboga, D. (2010). Artificial bee colony algorithm. *Scholarpedia*, 6915.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, *39*(3), 459–471. https://doi.org/10.1007/s10898-007-9149-x

Karaboga. (2005). An idea based on honeybee swarm for numerica optimization. *Technical Report TR06*.

Karsai, I., & Theraulaz, G. (1995). Nest building in a social wasp: postures and constraints. *Sociobiology, 26*, 83–114.

Katila, R., & Ahuja, G. (2002). Something Old, Something New: a Longitudinal Study of Search Behavior and New Product Introduction. *Academy of Management Journal*, *45*(6), 1183–1194. https://doi.org/10.2307/3069433

Kauffman, M., & Levin, S. (1987). Towards a general theory of adaptive walks on rugged landscapes. *Journal of theoretical Biology, 128*(1), 11-45.

Kauffman, S. A. (1989). The NK model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of theoretical biology, 141*(2), 211-245.

Kauffman, S. A., & Weinberger, E. D. (1989). The NK model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of theoretical biology, 141*, 211-245.

Kennedy, & Eberhart. (1995). Particle swarm optimization. *IEEE International Conference on Neural Networks*, (p. 1942–1948).

Kennedy, J. (2011). Particle swarm optimization. In *Encyclopedia of machine learning* (p. 760-766). Springer.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Neural Networks, 1995. Proceedings., IEEE International Conference on*, *4*, 1942–1948 vol.4. https://doi.org/10.1109/ICNN.1995.488968

Kerr, N., & Tindale, R. (2004). Group performance and decision making. *Annu. Rev. Psychol.*, 623–655.

Kinouchi, O., & Copelli, M. (2006). Optimal dynamical range of excitable networks at criticality. *Nature Physics*, *2*(5), 348–351. https://doi.org/10.1038/nphys289

Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*, 671-680.

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., & Porter, M. A. (2014). Multilayer networks. *Journal of Complex Networks*, *2*(3), 203–271. https://doi.org/10.1093/comnet/cnu016

Kozlowski, S. W., and Bell B. S., Work groups and teams in organizations, *Handbook of psychology* (2003).

Krause, J., Ruxton, G. D., & Krause, S. (2010). Swarm intelligence in animals and humans. *Trends in ecology & evolution, 25*(1), 28-34.

Krieger, M. J., Billeter, J. B., & Keller, L. (2000). Ant-like task allocation and recruitment in cooperative robots. *Nature, 406*(6799), 992-995.

Krihnanand, K., & Ghose, D. (2009). Glowworm Swarm Optimization for Simultaneous Capture of Multiple Local Optima of Multimodal Functions. *Journal of Swarm Intelligence*, 87–124.

Krihnanand, K., & Ghose, D. (2009). Glowworm Swarm Optimization: A new method for optimizing multi-modal function. *Journal of Computational Intelligence Studies*, 93–119.

Langer, J. S. (1980). Instabilities and pattern formation in crystal growth. *Reviews of Modern Physics, 52*(1), 1.

Laughlin, P. R. (2003). Groups perform better than the best individuals on letters-to-numbers problems: Informative equations and effective strategies. *Journal of Personality and Social Psychology, 85*(4), 684.

Laughlin, P. R. (2006). Groups perform better than the best individuals on letters-to-numbers problems: effects of group size. *Journal of Personality and social Psychology, 90*(4), 644.

Lawler, E.E. III, Mohrman S.A., and Ledford, G.E. Jr., Creating high performance organizations: Practices and results of employee involvement and total quality management in Fortune 1000 companies (San Francisco: Jossey-Bass, 1995).

Lee, C. H. (2014). Simple model for multiple-choice collective decision making. *Physical Review E, 90*(5), 052804.

Lee, K. M., Min, B., & Goh, K. Il. (2015). Towards real-world complexity: an introduction to multiplex networks. *European Physical Journal B*, *88*(2). https://doi.org/10.1140/epjb/e2015-50742-1

Leone, M., Vázquez, A., Vespignani, A., & Zecchina, R. (2002). Ferromagnetic ordering in graphs with arbitrary degree distribution. *The European Physical Journal B - Condensed Matter and Complex Systems, 28*(2), 191-197.

Lepine, J. A., Piccolo, R. F., Jackson, C. L., Mathieu, J. E., and Saul, J. R., A meta-analysis of teamwork processes: tests of a multidimensional model and relationships with team effectiveness criteria, *Personnel Psychology* **61** (2008) 273–307.

Levinthal, D. A. (1997). Adaptation on rugged landscapes. *Management Science*, *43*(7), 934–950. https://doi.org/10.1287/mnsc.43.7.934

Li, X. L. (2002). An optimizing method based on autonomous animats: fish-swarm algorithm. *System Engineering Theory and Practice, 22*(11), 32-38.

Loch, C., Mihm, J., & Huchzermeier, A. (2003). Concurrent Engineering and Design Oscillations in Complex Engineering Projects. *Concurrent Engineering*, *11*(3), 187–199. https://doi.org/10.1177/106329303038030

Lynn, G. S., Reilly, R. R., and Akgün, A. E., Knowledge management in new product teams: practices and outcomes, *Engineering Management, IEEE Transactions on* **47**(2) (2000) 221-231.

Malone, T. W., & Bernstein, M. S. (2015). *Handbook of collective intelligence.* MIT Press.

Mathieu, J. e. (2008). Team effectiveness 1997-2007: a review of recent advancements and a glimpse into the future. *J. Manag. Stud., 34*, 410–476.

Meinhardt, H., & Meinhardt, H. (1982). *Models of biological pattern formation* (Vol. 6). London: Academic Press.

Melouk, S., Damodaran, P., & Chang, P.-Y. (2004). Minimizing makespan for single machine batch processing with non-identical job sizes using simulated annealing. *International Journal of Production Economics, 87*, 141-147.

Metropolis, M., Rosenbluth, A., Rosenbluth, M., Teller, A., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics, 21*, 1087-1092.

Mirjalili, S. (2015). The ant lion optimizer. *Advances in Engineering Software, 83*, 80-98.

Moncayo-Martinez, L. A., & Zhang, D. Z. (2011). Multi-objective ant colony optimisation: A meta-heuristic approach to supply chain design. *International Journal of Production Economics, 131*, 407-420.

Mora, T. &. (2011). Are biological systems poised at criticality? *Journal of Statistical Physics, 144*(2), 268-302.

Newman, M.E.J., Networks. An Introduction, (Oxford: Oxford University Press, 2010).

Nia, A. R., Far, M. H., & Niaki, S. T. (2014). A fuzzy vendor managed inventory of multi-item economic order quantity model under shortage: An ant colony optimization algorithm. *International Journal of Production Economics, 155*, 259-271.

Onsager, L. (1944). Crystal statistics. I. A two-dimensional model with an order-disorder transition. *Physical Review, 65*(3-4), 117.

Ottino, J. M. (2004). Engineering complex systems. *Nature, 427*, 399-399.

Parrish, J. K., Viscido, S. V., & Grünbaum, D. (2002). Self-organized fish schools: an examination of emergent properties. *Biological Bulletin, 202*(3), 296–305.

Pelled, L. H., Eisenhardt, K. M., and Xin, K. R., Exploring the black box: An analysis of work group diversity, conflict and performance, *Administrative science quarterly* **44**(1) (1999) 1-28.

Pérez-Escudero, A., & de Polavieja, G. G. (2011). Collective animal behavior from bayesian estimation and probability matching. *PLoS Computational Biology*, *7*(11). https://doi.org/10.1371/journal.pcbi.1002282

Poli, Kennedy, & Blackwell. (2007). Particle Swarm Optimization an Overview. *Swarm Intelligence*, 1–25.

Price, K., Storn, R., & Lampinen, J. (2005). *Differential Evolution: A Practical Approach to Global Optimization.* Springer.

Radhakrishnan, S., & Ventura, J. A. (2000). Simulated annealing for parallel machine scheduling with earliness-tardiness penalties and sequence-dependent set-up times. *International Journal of Production Research, 38*, 2233-2252.

Raja, P., & Bhaskaran, V. (2013). Improving the Performance of Genetic Algorithm by reducing the population. *International Journal of Emerging Technology and Advanced Engineering*, 86–91.

Reynolds, C. W. (1987). Flocks, herds and school: a distributed behavioral model. *Computer Graphic, 21*(4), 25–34.

Romeo, F., & Sangiovanni-Vincentelli, A. (1991). A theoretical framework for simulated annealing. *Algorithmica, 6*, 302-345.

Rubenstein, M. C. (2014). Programmable self-assembly in a thousand-robot swarm. *Science, 345*(6198), 795-799.

Rubenstein, M., Cornejo, A., & Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. *Science, 345*(6198), 795-799.

Şahin E., & Winfield, A. (2008). Special issue on swarm robotics. *Swarm Intelligence, 2*, 69–72.

Şahin, E. (2005). Swarm robotics: from sources of inspiration to domains of application. *Swarm Robotics Workshop: State of the Art Survey*, (p. 10–20). Berlin.

Sanna, L. J., and Parks, C. D., Group research trends in social and organizational psychology: Whatever happened to intragroup research?, *Psychological Science* **8** (1997) 261-267.

Schweitzer, F. &. (2007). *Brownian agents and active particles: collective dynamics in the natural and social sciences.* Springer Science & Business Media.

Seneta, E. (1981). *Chains, Non-negative Matrices and Markov.* New York: Springer.

Seyfried, J., Szymanski, M., Bender, N., Estana, R., Thiel, M., & Worn, H. (2005). The i-swarm project: intelligent small world autonomous robots for micro-manipulation. *Swarm Robotics Workshop: State-of-the-Art Survey. 3342*, p. 70–83. Berlin, Germany: Springer.

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. *IEEE International Conference on Evolutionary Computation*, (p. 69–73).

Sornette, D. (2014). Physics and financial economics (1776–2014): puzzles, Ising and agent-based models. *Reports on Progress in Physics, 77*(6), 062001.

Soysal, O., & Sahin, E. (2005). Probabilistic aggregation strategies in swarm robotic systems. *IEEE Swarm Intelligence Symposium*, p. 325-332.

Stauffer, D., Social applications of two-dimensional Ising models, *American Journal of Physics*, **76**(4) (2008) 470-473.

Storn, R., & Price, K. (1997). Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 341–359.

Stützle, T., & Hoos, H. (2000). MAX-MIN Ant System. *Future Generation Computer System, 16*, 889–914.

Sumpter, D. J. T., & Pratt, S. C. (2009). Quorum responses and consensus decision making. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, *364*(1518), 743–53. https://doi.org/10.1098/rstb.2008.0204

Surowiecki, J. (2004). *The Wisdom of the Crowds: Why the Many are Smarter than the Few.* Little Brown.

Tatikonda, M. V., and Montoya-Weiss, M. M., Integrating operations and marketing perspectives of product innovation: The influence of organizational process factors and capabilities on development performance, *Management Science* **47**(1) (2001) 151-172.

Teodorovic, D., Lucic, P., Markovic, G., & Dell'Orco, M. (2006, September). Bee colony optimization: principles and applications. In 2006 *8th Seminar on Neural Network Applications in Electrical Engineering* (pp. 151-156). IEEE.

Theraulaz, G. B. (1998). The origin of nest complexity in social insects. *Complexity,, 3*(6), 15–25.

Thorpe, W. H. (1963). Learning and instinct in animals. London: Methuen.

Tosuna, U., Dokeroglua, T., & Cosara, A. (2013). A robust Island Parallel Genetic Algorithm for the Quadratic Assignment. *International Journal of Production Research, 51*(14), 4117–4133.

Traniello, J. F., & Robson, S. K. (1995). *The chemical ecology of insects.* London: W. J. Bell & R. Cardé.

Trianni, V., Groß, R., Labella, T. H., Şahin, E., & Dorigo, M. (2003, September). Evolving aggregation behaviors in a swarm of robots. *European Conference on Artificial Life*, p. 865-874.

Tschinkel, W. R. (2003). Subterranean ant nests: trace fossils past and future? *Palaeogeography, Palaeoclimatology,Palaeoecology, 192*(1–4), 321–333.

Turalska, M., & West, B. J. (2014). Imitation versus payoff: Duality of the decision-making process demonstrates criticality and consensus formation. *Physical Review E*, 90(5), 052815.

Turalska, M., Lukovic, M., West, B. J., & Grigolini, P. (2009). Complexity and synchronization. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, *80*(2). https://doi.org/10.1103/PhysRevE.80.021110

Turalska, M., West, B. J., & Grigolini, P. (2011). Temporal complexity of the order parameter at the phase transition. *Physical Review E*, *83*(6), 061142.

Üçoluk, G. (2002). Genetic algorithm solution of the TSP avoiding special crossover and mutation. *Intelligent Automation & Soft Computing, 8*, 265–272.

Valdez, F., & Chaparro, I. (2013). Ant Colony Optimization for solving the TSP symmetric with parallel processing. *Joint IFSA World Congress and NAFIPS Annual Meeting*, 1192–1196.

Vanni, F., Luković, M., & Grigolini, P. (2011). Criticality and transmission of information in a swarm of cooperative units. *Physical review letters, 107*(7), 078103.

Vicsek, T., & Zafeiris, A. (2012). Collective motion. *Physics Reports*. https://doi.org/10.1016/j.physrep.2012.03.004

Vicsek, T., Czirk, A., Ben-Jacob, E., Cohen, I., & Shochet, O. (1995). Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, *75*(6), 1226–1229. https://doi.org/10.1103/PhysRevLett.75.1226

Vittori, K., Talbot, G., Gautrais, J., Fourcassié, V., Araujo, A. F., & Theraulaz, G. (2006). Path efficiency. *Journal of Theoretical Biology, 239*, 507–515.

W. M. Spears, D. F. (2004). Distributed, physics-based control of swarms of vehicles. *Autonomous Robots, 17*, 137–162.

Wang, Z., Szolnoki, A., & Perc, M. (2013). Interdependent network reciprocity in evolutionary games. *arXiv preprint arXiv:1308.1947*.

Wang, Z., Szolnoki, A., & Perc, M. (2013). Optimal interdependence between networks for the evolution of cooperation. *Scientific Reports*, *3*, 2470. https://doi.org/10.1038/srep02470

Wang, Z., Wang, L., Szolnoki, A., & Perc, M. (2015). Evolutionary games on multilayer networks: a colloquium. *European Physical Journal B*, *88*(5). https://doi.org/10.1140/epjb/e2015-60270-7

Ward, A. J. W., Herbert-read, J. E., Sumpter, D. J. T., Krause, J., Ward, A. J. W., & Herbert-read, J. E. (2011). Fast and accurate decisions through collective vigilance in fish shoals. *Proceedings of the National Academy of Sciences of the United States of America*, *108*, 2312–5. https://doi.org/10.1073/pnas.1007102108

Ward, A. J. W., Sumpter, D. J. T., Couzin, I. D., Hart, P. J. B., & Krause, J. (2008). Quorum decision-making facilitates information transfer in fish shoals. *Proceedings of the National Academy of Sciences of the United States of America*, *105*(19), 6948–6953. https://doi.org/10.1073/pnas.0710344105

Watts, D. J. (2002). A Simple Model of Global Cascades on Random Networks. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(9), 5766–5771. https://doi.org/10.2307/3058573

Weidlich, W. (1971). THE STATISTICAL DESCRIPTION OF POLARIZATION PHENOMENA IN SOCIETY†. *British Journal of Mathematical and Statistical Psychology, 24*(2), 251-266.

Weidlich, W. (1991). Physics and social science—the approach of synergetics. *Physics Reports, 204*(1), 1-163.

Weinberger, E. D. (1996). Np completeness of kauffman's nk model, a tuneable rugged fitness landscape. *Santa Fe Institute working paper*. Tratto da http://www.santafe.edu/media/workingpapers/96-02-003.pdf.

Werfel, J. P. (2014). Designing collective behavior in a termite-inspired robot construction team. *Science, 343*(6172), 754-758.

Wilson, E. O. (1971). *The insect societies.* Cambridge:: Harvard University Press.

Winfield, A., Harper, C., & Nembrini, J. (2005). Towards dependable swarms and a new discipline of swarm engineering. *Swarm Robotics Workshop: State-of-the-Art Survey* (p. 126-142). Berlin,: Şahin and Spears.

Wolfers, J., & Zitzewitz, E. (2004). Prediction markets. *J. Econ. Persp., 18*, 107–126.

Wolpert, D. H., & M. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*, 67–82.

Woolley, A. W., Chabris, C. F., Pentland, A., Hashmi, N., & Malone, T. W. (2010). Evidence for a collective intelligence factor in the performance of human groups. *science, 330*, 686-688.

Wu, Y., Lee, W., & Chien, C. (2011). Modified the Performance of Differential Evolution Algorithm with Dual Evolution Strategy. *International Conference on Machine Learning and Computing*, (p. 57–63).

Yagmahan, & Yenisey. (2010). A multi-objective ant colony system algorithm for flow shop scheduling. *Expert Systems with Applications*, 1361–1368.

Yan, Wu, Liu, & Huang. (2013). An Improved Particle Swarm Optimization Algorithm and Its Application. *International Journal of Computer Science*, 316–324.

Yang, X.-S., & Deb, S. (2010). Engineering optimisation by cuckoo search. *International Journal of Mathematical Modelling and Numerical Optimisation, 1*, 330-343.

Zainal, N., Zain, A., Radzi, N., & Udin, A. (2013). Glowworm Swarm Optimization (GSO) Algorithm for Optimization Problems: A State-of-the-Art Review. *Applied Mechanics and Materials, 421*, 507–511.

Zhang, Y., Ma, X., Gu, Y., & Miao, Y. (2011). A modified glowworm swarm optimization for multimodal functions. *Chinese Control and Decision Conference*, (p. 2070–2075).