2nd Symposium on Nature-Inspired Computing and Applications (NICA)

Ed Keedwell (editor)

AISB Convention 2013 • University of Exeter • 3rd–5th April, 2013

Foreword from the Convention Chairs

This volume forms the proceedings of one of eight co-located symposia held at the AISB Convention 2013 that took place 3rd-5th April 2013 at the University of Exeter, UK. The convention consisted of these symposia together in four parallel tracks with five plenary talks; all papers other than the plenaries were given as talks within the symposia. This symposium-based format, which has been the standard for AISB conventions for many years, encourages collaboration and discussion among a wide variety of disciplines. Although each symposium is self contained, the convention as a whole represents a diverse array of topics from philosophy, psychology, computer science and cognitive science under the common umbrella of artificial intelligence and the simulation of behaviour.

We would like to thank the symposium organisers and their programme committees for their hard work in publicising their symposium, attracting and reviewing submissions and compiling this volume. Without these interesting, high quality symposia the convention would not be possible.

Dr Ed Keedwell & Prof. Richard Everson
AISB 2013 Convention Chairs

Published by
The Society for the Study of Artificial Intelligence and the Simulation of Behaviour
http://www.aisb.org.uk

ISBN: 978-1-908187-37-6

2nd Symposium on Nature Inspired Computation and Applications (NICA)

Preface

This second symposium on Nature Inspired Computation and Applications (NICA) builds on the success of the first NICA symposium held at AISB 2012. The NICA symposium series takes inspiration from previous successful symposia that focussed on specific algorithms, including evolutionary algorithms and swarm intelligence. The papers in this symposium demonstrate the breadth of research in this topic in AI, covering spiking neural networks, biogeography based methods and multi-agent systems applied to problems in engineering, AI research and ecology. It is clear from these papers that the field of Nature Inspired Computation encompasses a wide variety of techniques that can be used as tools not only for optimising systems but also for understanding our own cognitive processes.

I would like to thank the authors who submitted papers to the symposium and to the NICA 2013 programme committee (listed below) for their assistance in reviewing submissions this year.

Prof. Richard Everson
Dr Jonathan Fieldsend
Dr Antony Galton
Dr Kent McClymont
Dr David Walker

Dr Ed Keedwell, Symposium Chair

Table of Contents

Magnitron: A spiking neural network model of numerical cognition

Philippe Chassy,
Department of Psychology, Liverpool Hope University, Hope Park, Liverpool, UK, L16 9JD, chassyp@hope.ac.uk

Neil Buckley, David Reid
Department of Computer Sciences, Liverpool Hope University, Hope Park, Liverpool, UK,
L16 9JD,

Abstract. The authors present an artificial neural network model of human perception of natural numbers. Evidence from experimental psychology and neuroimaging data has been used to set the parameters and the structure of the model. A new architecture, *Magnitron*, has been designed to implement how human agents perceive magnitudes both in analogue format with the inherited system and in symbolic format with the culturally-recycled system. The results demonstrate the model's ability to simulate human perception of quantities in both analogue and symbolic format. The model presented in this paper is, at the best of our knowledge, the first spiking neural network model of numerical cognition and thus offers a new avenue for all scientists interested in numerical cognition.

1 INTRODUCTION

One of the greatest achievements of mankind has been to put a man on the moon. Without millenaries of progress in mathematics such performance, like many others, would not have been possible. At the core of all mathematical abilities and intuitions lies the notion of quantity [8]. Neuroimaging evidence [13] has shown that numerical cognition is taking place in the intraparietal sulci (IPS). Yet, only recently did scientists explore how the numerical value is extracted from the stimulus. The process whereby the information from the visual display on the retina is turned into a quantity in the IPS has been termed number mining [4].

1.1. Inherited and cultural numerical cognition

Natural numbers can be coded in two different formats: analogue and symbolic. Whereas the analogue format implies that the quantity is a perceptually accessible feature (e.g., four dots:❖), the symbolic format implies an arbitrary relation between symbol and quantity (e.g., 4). Though trivial at first sight, the difference between the two systems reflects the evolution of numerical cognition from animals to humans.

Perceptual discrimination of quantities in analogue format is surprisingly efficient. It has soon been shown that animals are able to process natural numbers [17]. In parallel, developmental and anthropologic evidence demonstrates that infants and adults with no training in mathematics are able to perform well with small natural numbers [11, 16]. The processing of small quantities leads to similar performance in infants, non-educated adults and animals, suggesting that the analogue system is shared across species [13]. The symbolic format is the result of enculturation. In most societies, the first training in mathematics comes when individuals learn to pair symbols with small natural numbers. Such a symbolic format increases our capacity to represent and manipulate magnitudes. Dehaene and Cohen have put forward the idea that additional neural networks have been recruited during the course of evolution to cope with language demands [7]; in other words the visual system has evolved to detect quantities.

1.2. Number mining

The way in which the value is extracted from the analogue/inherited system and the cultural/symbolic system has been the focus of Chassy and Grood's [4] study. They used fMRI to map the neural pathways that carry out number mining in analogue and symbolic format.

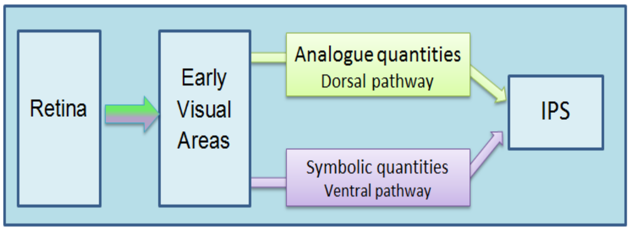

Figure 1 - Two neural paths for number mining

Referring to Figure 1, from the retina to early visual areas the neural signals takes the same route for both types of format. There a hub, in charge of detecting the nature of the signal, forwards it to either the symbolic or the analogue route. Both routes process number mining and forward their result to the third subdivision of the right IPS (termed right hIP3, see [4]), wherein neurons code exact quantities such as natural numbers. Number mining is processed by partly different networks; the difference reflecting the evolution of our visual system.

1.3. Neural constrains on the computer implementation

The fact that both systems are served by neural routes that partly differ suggests that learning effects are autonomous to each route. Neural plasticity, the mechanisms whereby biological neural networks are reorganized to store memories [12], would take place independently in the ventral and dorsal pathways. Therefore, number mining should be implemented as two different neural networks. Both analogue and symbolic systems have in common the network between the retina and the neural hub. Accordingly, the same spiking neural network will be used to implement common early visual processing. The hub has been designed as a spiking neural network (SNN) that detects the format of the input and adequately routes the signal to the appropriate path. Recent research suggests that the neural hub is located in visual area 3 (i.e., hOC3v) [5].

The implementation of the IPS is more challenging. Due to the limit in spatial resolution of fMRI measurements, the anatomy of IPS is not well known. The inner structure of the hIP3 subsection is not understood to a fine degree of detail and thus it is impossible at the moment to implement the processing inside the IPS. For the present work we assumed that the quantity held in hIP3 is coded as a cell assembly [10, 17]. One layer of neurons will implement units and one output neuron will represent the final quantity.

2 METHOD

2.1 Architecture

Our model simulates quantity perception along the two routes and its final representation in the IPS. It is termed *Magnitron* (standing for electronic perception of magnitudes) and its structure, presented in Figure 2, reflects current knowledge on number mining. The model is made of five components: the retinal input, the neural hub, the two number mining pathways and the final representation in the IPS.

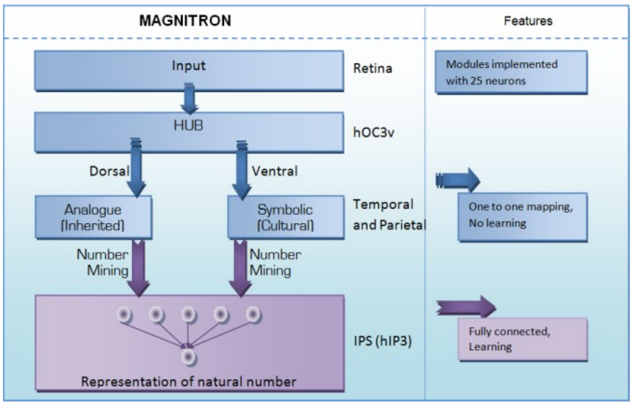

Figure 2 - The architecture of *Magnitron*

The input layer of *Magnitron* is composed of a matrix of 5*5 cells that encode the visual field. The input is a simplified version of the retina. Since the experiments carried out in numerical magnitude do not manipulate colour we considered that the input layer would only implement the retina's sensitivity to light. The hub is a neural network determining whether the input is in analogue or digital format.

To reflect the dichotomy between the neural networks performing magnitude estimation in analogue and digital format, we designed two separate SNNs. *Magnitron*$_{(A)}$ is the SNN carrying out number mining in the analogue format and *Magnitron*$_{(S)}$ is the SNN carrying out number mining in the symbolic format. Both neural networks had the same architecture: a 25 element input and a 5-element output. The weights connecting the two layers reflected the neural connections that are in charge of performing number mining. Both networks directly connect to the IPS module. The output made of five neurons represents the neurons in the IPS that are coding quantities; the sum of these was then fed into the final neuron implementing the ability to represent a natural number that is accessible to consciousness.

To illustrate the computational power of spiking neural networks we also designed a comparable traditional neural network model of the analogue and symbolic number mining pathways. These networks were made of three layers of sigmoid neurons. For the comparison to be compatible with the Retina, the input was made of 25 neurons. Similarly, the output reflected the entry of the IPS and thus was made of 5 neurons. The hidden layer was made of 10 neurons. Two Feedforward neural networks were thus designed to implement the two number mining pathways. In total, we thus designed two models of number mining, each implementing the two number mining pathways:

(1) A Feedforward neural network model: FN$_{(A)}$ and FN$_{(S)}$.
(2) A SNNs model : *Magnitron*$_{(A)}$ and *Magnitron*$_{(S)}$.

2.2 Neural level

At the neural level, we implemented a signal that is close to the biological reality. We used the integrate-and-fire neuron, which allowed an implementation that was less demanding on resources than Hodgkin-Huxley neurons [14] but reflects the neurons' ability to process time and space integration of the inputs. Similarly to previous models of neural activity [1-2], we have implemented the connection between two neurons with several synaptic terminals. Each connection can be regarded as a "sub-synapse" with its own transmission delay and efficacy. Learning, actual neural plasticity at the cell level [12], was implemented with Booij's version of the backpropagation algorithm [3]. Equation 1 provides the level of activity of a neuron at every moment and Equation 2 shows the calculation of the correction of a weight in the SNNs.

Eq(1) Activity:

$$v_j = \sum_{i \in \Gamma_j, s_i \in F_i, d^k \in D_{ij}} w_{ij}\, \epsilon(t - s_i - d^k) \;-\; \sum_{s_i \in F_i} \eta(t - s_i)$$

where Γ_j is the set of neurons pre-synapstic to neuron j, F_i is the set of firing times of neuron i, D_{ij} is the set of delay values between neurons i, d^k is the kth delay value in that set, t is the current simulation time and ε(x) is an exponential function approximating biological spike response.

$$\Delta w_{ji}^k = -\eta \frac{-\sum_{t_i^{(g)} \in F_i} \epsilon\left(t_j^{(1)} - t_i^{(g)} - d^k\right)}{\sum_{i,k} \sum_{t_i^{(g)} \in F_i} w_{ij}^k \epsilon'\left(t_j^{(1)} - t_i^{(g)} - d^k\right)} (t_j^{(1)} - s_j^{(1)})$$

Eq(2) learning:

where Δw_{ji}^k is the weight change of the kth sub-synapse between neurons i and j, η is the learning rate, $t_i^{(g)}$ is the gth spike time of neuron i and $t_j^{(1)}$ is the first spike time of neuron j.

2.3 Stimuli

The stimuli were a digitalised version of those used in fMRI experiments [4, 9]. Their relevance to simulate human performance and their credibility as ecologically valid stimuli is thus ensured. In Figure 3, the left panel shows four examples of analogue stimuli wherein the natural numbers are encoded as sets of dots. The right panel shows the corresponding numbers in symbolic format. We have limited our simulation to numerals up to five since both behavioural and neural evidence suggest a specific, inherited capacity to process these quantities [15]. To be fed in *Magnitron*, the stimuli were converted into a one-dimensional vector.

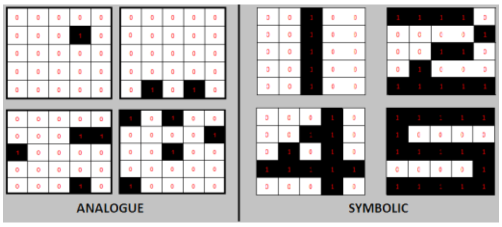

Figure 3 - Examples of analogue and symbolic stimuli

2.4 Training algorithms

The input display is coded as a spike train into the input neurons. Encoding of the spike train is by constructing (relative) early and late spikes (resp. 0 and 20ms for resp. white and black pixels).
The encoded input is selected from a shuffled set of values. The spikes then propagate through each input synapse according to the concomitant sub-synaptic delays. The total membrane potential, v_i, arriving at each hidden layer neuron, i, is therefore given in Equation 3:

$$\text{Eq(3):} \quad V_i = \sum_{h=1}^{|\Gamma|} w_{hi}^{k} \epsilon(t - s - d_{hi}^{k})$$

where the weight is set at a constant 1.5 for this input layer only. If $v_i > \theta$ (where θ=threshold) then i spikes and likewise transmits m spikes to each output neuron j. At the points when $v_j > \theta$, j spikes, and its time-to-first-spike is taken as that neuron's output. An early (resp. late) output spike time, $t_j^{(1)}$, is decoded as binary 0 (resp. 1). The output layer thus forms a binary vector, whose Hamming weight is taken as the integer output of the SNN. Using spike times of all hidden and output neurons, Eq(3) is used to calculate the weight change of each respective sub-synapse. The SNN then continues to the next training epoch.
For each epoch, the Hamming weight of the output vector is subtracted from the expected output to give the error for the present epoch, and the mean squared error (MSE) is calculated for the previous n epochs, where n is the number of training sets. The simulation runs for 1000 epochs. Each time a new global minimum MSE is found the state of the SNN is saved to a file. The final saved SNN is the optimum solution.

3 RESULTS

The performance of the hub in classifying inputs as analogue or symbolic was 100%. Hence, the results will focus on the actual performance of the networks implementing the pathways.

3.1 Ideal visual conditions

3.1.1 *Analogue system*

The two neural networks, FN and *Magnitron*, were trained as indicated in §2.4 and tested with 10 sets of 5 stimuli. FN performed at 68.00% (SD = 16.87%) with increasing error when the number of dots in the set increased. In comparison, after training (see figure 4), *Magnitron* performed at 76.00% (SD = 18.38). The difference in performance between the two networks was not statistically significant, $t(18) = -1.01$, $p = .32$.

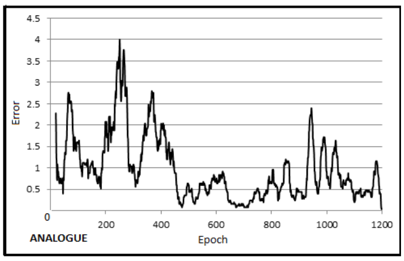

Figure 4 - Error as a function of epoch

3.1.2 *Symbolic system*

The two neural networks, FN and *Magnitron*, were trained as indicated in §2.4 and tested with 2 sets of 5 stimuli. FN performed at 100% with increasing error when the number of dots in the set increased. In comparison, after training (see figure 4), *Magnitron* performed at 100%.

3.2 Testing flexibility

The perfect performance of *Magnitron* with a reduced set of symbols has led us to test *Magnitron*'s ability in different visual conditions. To implement the case wherein objects are partly occluded, we altered the quality of the image by deleting some bits in each display. Also, research in reading has indicated that better readers tend to centre the gaze on a specific place of the word. To reflect such form of expertise we shifted some displays to the left or to the right and tested *Magnitron*'s ability to classify this data. The network was then trained with a new set of stimuli (See figure 5) and then tested.

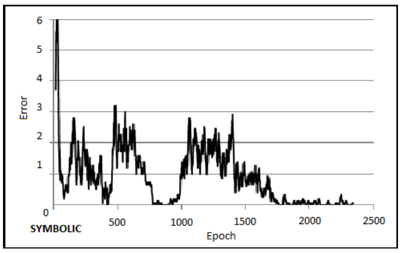

Figure 5 - Error as a function of epoch

After training, the network was asked to classify 5 sets of digits that displayed either distortions or shifts. The network performance dramatically dropped to an average of $M = 80.00$ % ($SD = 14.14$).

4 DISCUSSION

In this work we attempted to simulate human numerical cognition with a SNN-based architecture termed *Magnitron*. *Magnitron*'s performance was tested with information coded in both analogue and symbolic formats and then compared to performance of human agents and feedforward neural networks.
Magnitron did not perform at 100% with analogue stimuli, a level of performance that human agents reach easily. The performance, though disappointing, is indicative of the complexity of the human visual system. It further suggests that

the neural networks that have evolved to support the counting of elements in a visual display carry out a sophisticated operation.

In stark contrast, *Magnitron* displays perfect performance with centred symbols. This result points to the superiority of symbols over non-formalised displays and thus supports the point of view that language has facilitated the development of mathematical abilities. We have shown that performance drops by as much as 20% if the stimuli are altered or shifted. Even though *Magnitron* retains some discriminatory power when the 5*5 retina is not centred on the target digit, we consider 80% a correct performance for a first implementation. The centring of an image on the retina is due to occulo-motor corrections; the implementation of which is out of the scope of *Magnitron*.

When comparing the performance of *Magnitron* in processing analogue and symbolic displays, our work suggest the counterintuitive result that the symbolic, culturally-based, coding of natural numbers is more easily implemented. The work also shows that the *Magnitron* performs better than FN for analogue stimuli. We interpret such a result as evidence that symbolic processing is less metabolically demanding that analogue processing, an interpretation that is also consistent with fMRI findings [4].

Beyond the limits of the implementation of the analogue mining network, the implementation of the IPS might also require improvements. Most of the scanners performing fMRI research are not sufficiently powerful (e.g., 1.5-3T) to investigate low level anatomical details. We were thus left to decide the structure of the IPS without much information. The new generation of powerful scanners (e.g., 9.4T) will shed a new light on the functional neuroanatomy of the horizontal segment of the IPS. With respect to the information processing at the neurons level, it has been suggested that they respond as ‘Gaussian’ detectors [6], a notion that was not compatible with the architecture of the networks (notably the FN). The combination of detailed fMRI research and modelling of mathematical cognition will undoubtedly lead us to a more sophisticated implementation of information processing in the IPS.

5 CONCLUSION

Our model has shown that numerical magnitudes can be implemented with a spiking neural network. Crucially, we managed to implement successfully the analogue and the symbolic route. The results have revealed that the neural machinery performing number mining in analogue format should be more sophisticated. Though the average performance of *Magnitron* is inferior to other models [18], it also is much closer to the biological reality and realizes the first step towards a biologically-inspired model of numerical cognition.

6 ACKNOWLEDGEMENTS

We thank Dr Hissam Tawfik for support in the project and two anonymous reviewers for their helpful comments.

7 REFERENCES

[1] Bohte, S., Kok, J.N., Poutre, H.L., (2002) "Spikeprop: Error-backpropagation for in multi-layer networks of spiking neurons". *Neurocomputing*, **48**, pp. 17-37.

[2] Bohte, S.M., Kok, J.N., La Poutré, H. *Spikeprop: Backpropagation for networks of spiking neurons*. in *ESANN 2000*. 2000.

[3] Booij, O., *Temporal pattern classification using spiking neural networks*. 2004, Universiteit van Amsterdam: Amsterdam, The Netherlands.

[4] Chassy, P., Grodd, W., (2012) "Comparison of quantities: Core and format-dependent regions as revealed by fMRI". *Cerebral Cortex*, **22**, No 6. pp. 1420-1430.

[5] Chassy, P., Hösl, I., Grodd, W., *Neural codes of number words: An fMRI exploration*, in *Annual Conference of the British Psychological Society*. 2012: London, UK.

[6] Dehaene, S., *Symbols and quantities in parietal cortex: Elements of a mathematical theory of number representation and manipulation*, in *Sensorimotor foundations of higher cognition*, P. Haggard, Y. Rossetti, and M. Kawato, Editors. 2007: Harvard University Press. p. 527-574.

[7] Dehaene, S., Cohen, L., (2007) "Cultural recycling of cortical maps". *Neuron*, **56**, No. 2 pp. 384-398.

[8] Dehaene, S., (2009) "Origins of mathematical intuitions". *The Year in Cognitive Neuroscience*, **1156**, pp. 232-259.

[9] Fias, W., Lammertyn, J., Reynvoet, B., Dupont, P., Orban, G.A., (2003) "Parietal representation of symbolic and nonsymbolic magnitude". *Journal of Cognitive Neuroscience*, **15**, pp. 47–56.

[10] Hebb, D.O., *Organization of behavior*. 1949, New Jersey: Wiley and Sons.

[11] Izard, C., Sann, C., Spelke, E., Streri, A., (2009) "Newborn infants perceive abstract numbers". *Proceeding of the National Academy of Science USA*, **106**, No 25. pp. 10382–10385.

[12] Kandel, E.R., (2001) "The molecular biology of memory storage: A dialogue between genes and synapses". *Science*, **294**, pp. 1030-1038.

[13] Nieder, A., Dehaene, S., (2009) "Representation of number in the brain". *Annual Review of Neuroscience*, **32**, No. pp. 185-208.

[14] Paugam-Moisy, H., Bohte, S., *Computing with spiking neural networks*, in *Handbook of natural computing*, T. Rozenberg, T. Back, and J. Kok, Editors. 2009, Springer-Verlag. p. 1-47.

[15] Piazza, M., (2010) "Neurocognitive start-up tools for symbolic number representation". *Trends in Cognitive Sciences*, **14**, No 12. pp. 542-551.

[16] Pica, P., Lemer, C., Izard, V., Dehaene, S., (2004) "Exact and approximate arithmetic in an amazonian indigene group". *Science*, **306**, pp. 499–503.

[17] Tudusciuc, O., Nieder, A., (2007) "Neuronal population coding of continuous and discrete quantity in the primate posterior parietal cortex". *Proceedings of the National Academy of Science USA*, **104**, pp. 14513–1418.

[18] Zorzi, M., Stoianov, I., Umilta, C., *Computational modelling of numerical cognition*, in *The handbook of mathematical cognition*, Campbell, Editor. 2005, Psychology Press.

Communication Success in Vague Category Games

Henrietta Eyre[1] and **Jonathan Lawry**

Abstract. The category game is a language game which models the communication protocols between two agents who aim to match category labels with objects encountered in a simulated or real world. We present a multi-agent category game in which category definitions explicitly incorporate semantic uncertainty and typicality. Furthermore, our model considers a mixed assertion set using a language where both basic labels and their negations are permitted. We argue that this conceptual framework is expressive and naturally generates robust assertion and concept updating models. Simulation results show that communication success increases to the highest value when a mixture of positive and negative assertions are used.

1 INTRODUCTION

In classical AI, communication between agents is often based on a fixed language and set of procedures. However within a complex and changing environment agents can better communicate and perform tasks if they can evolve their own semantic structures by adapting conceptual models depending on their interactions with other agents.

Steels has argued that in order to be more realistic, language models must take into account the evolutionary nature of language learning[16], and has proposed multi-agent systems as a useful research tool. In such systems, the syntactic and semantic structures may be individual to each agent, but groups of agents may then cooperate to evolve common structures and methods of communication. In [15] Steels defined a language game, building on the definition of Wittgenstein [19], to describe a full communicative interaction between two embodied agents. One agent, acting as a speaker, formulates a linguistic utterance to describe an object the environment. The speaking agent passes this utterance to the listening agent who then interprets its meaning and updates their conceptual model to satisfy any constraints implied by the assertion.

1.1 The Category Game

In this paper we investigate a formulation of the category game. The category game is language game which focuses on the set of all objects within a given attribute space. Category formation and mechanisms of categorisation are investigated by measuring the evolution of a system where agents have a set of communication rules and concepts which are updated through learning. Through the course of the game concepts emerge and evolve through basic communication between agents.

The category game was first proposed by Steels and Belpaeme [17], demonstrating that there are models according to which a population evolves so that agent concept definitions become sufficiently similar so as to allow for effective communication. As in [17], the category game is often performed on colour categories modelled on a continuous space (e.g. CIELAB or RBG). Such investigations can make use of the World Color Survey (WCS) [4], a catalog of language data from 110 unwritten languages. Berlin and Kay presented data from the WCS in support of the hypothesis that although different cultures use a different number of colour terms, the structure of colour categories is hierarchical, and colour categories demonstrate a cross-cultural universality [3]. Similar arguments have been presented using category game models. Work in [1] shows the emergence of universal colour categories in independent populations based on simulations using data from the WCS. Puglisi et al [13] have modelled the emergence of colour terms in a system modelling cultural input. Their results show remarkable agreement with the WCS. In addition, [6] shows the robustness for the evolutionary process of the language system under random noise. By applying a Bayesian inference procedure the system evolved to replicate typical patterns seen in modern languages.

1.2 Semantic Uncertainty

The category game typically uses a category model which may be mapped onto the unit interval, partitioned into discrete intervals for each concept. This approach results in a set of categories exclusive and exhaustive and does not allow for any explicit representation of uncertainty. An object cannot be represented by two categories, so a colour could not be both pink and red. Furthermore there cannot be an object which is not within the scope of any known category, and so richer linguistic structures such as label negations cannot be chosen as an appropriate communicative strategy. Both of these restrictions may seem counter-intuitive in some situations. Indeed vagueness is common in natural language. Situations where a lexical structure does not determine a unique interpretation may leave a collection of possible interpretations of meaning. In [18] Van Deemter provides a number of ways in which vagueness may be a positive feature of language. For instance, he suggests that risk may be minimised by using a vague assertions in the presence of conflict.

A conceptual model incorporating typicality was adopted in the category games literature [2] where the membership function of a category was taken to be the inverse Euclidean distance to the category prototype. This model was criticised in [13] on the grounds that you can define an equivalence relation between this and a rigid boundary model. In the following we introduce a category game model where concepts are modelled as uncertain regions of a metric space, providing agents with a conceptual model explicitly representing semantic uncertainty (the uncertainty associated with

---

[1] Department of Engineering Mathematics, University of Bristol, Merchant Venturers Building, Woodland Road, BS8 1UY, UK, email: henrietta.eyre@bristol.ac.uk,

the definition of concept labels resulting from the empirical manner in which we all learn to use language [10]) using a random set and prototype theory based neighbourhood model. Although uncertainty is modelled explicitly, this model is not subject to the criticism presented to [2] in [13].

2 MODELLING CONCEPTS

The classical theory of categorisation requires that all members of a given category satisfy a set of shared properties. Psychologists have criticised this model [14] arguing that a richer, more flexible representation is required for natural categories. The classical requirement that category membership be defined in terms of necessary and sufficient conditions renders the approach a highly inflexible process. Elements of the space which technically violate the category definition may in fact be very similar to category members.

2.1 Prototype Theory

Prototype theory is an alternative model of categorisation which was proposed by Rosch [14] and developed by Lakoff [9] and more recently Hampton [8]. Prototype theory demands that rather than satisfying a set of necessary and sufficient conditions, category members are categorised according to similarity to a prototypical member, or prototype. This model naturally results in a typicality ordering on category members.

2.2 Conceptual Spaces

A measure of conceptual distance is a necessary tool for prototype theory, and similarity spaces have been a topic of discussion over the last several decades [12, 14]. More recently Gärdenfors has proposed conceptual spaces [7] as a geometrical framework for concept representation. A conceptual space is a metric space where each dimension quantifies a certain property or feature, so that a given object is fully described by a point in this space. Categories within a conceptual space are represented as convex regions which may be generated by a Voronoi tessellation based around prototypical points. A Voronoi-based approach incorporating uncertainty in category boundaries was introduced in [5]. This model explicitly represents uncertain boundaries, but still provides a model which is exhaustive in its category definitions.

2.3 A Random Set and Prototype Neighbourhood Model

We use a neighbourhood concept model based on the label semantics framework [10, 11]. We define a conceptual space Ω and a finite set of n labels LA from which a compound set of expressions LE can be generated through recursive applications of logical connectives to labels in LA. Each label represents a word which may be used to represent elements of the conceptual space Ω, so that "x is L_i" is meaningful (although not necessarily appropriate) for any x in Ω. For example in CIELAB space, LA may be the labels "red", "blue" etc, and LE would be the expressions "red and blue", "not red" etc.

We assume a distance function d defined on Ω. A label L_i is then deemed to be appropriate to describe an element x if x is sufficiently close to the prototype for label L_i. More formally we say L_i is appropriate to describe x if $d(x, P_i) \leq \epsilon_i$ where ϵ_i is a distance threshold. We capture semantic uncertainty in ϵ_i, so ϵ_i is modelled as a random variable on $\mathbb{R}^+$.

Each agent is defined by an interpretation $I = (\Omega, d, \overrightarrow{P}, \overrightarrow{\epsilon})$ of LA, where $\overrightarrow{P}$ is a vector of prototypes and $\overrightarrow{\epsilon}$ is a vector of threshold variables. Given an interpretation, the constraint $d(x, P_i \leq \epsilon_i$ may be used to define a neighbourhood region of Ω within which all points may be appropriately used to describe x. We can define $\mathcal{N}^I_{L_i}$, the neighbourhood for L_i given interpretation I, and $\mathcal{N}^I_{\neg L_i}$, the neighbourhood for $\neg L_i$ given interpretation I, as follows:

$$\mathcal{N}^I_{L_i} = \{x : d(x, P_i) \leq \epsilon_i\} \tag{1}$$

$$\mathcal{N}^I_{\neg L_i} = (\mathcal{N}^I_{L_i})^c \tag{2}$$

The neighbourhood for a basic label L_i can be thought of as the closed ball with radius ϵ_i center which is a convex region with an uncertain boundary. Neighbourhoods may overlap with other neighbourhoods in the conceptual space, and there may be regions of the conceptual space where no neighbourhood is defined.

3 A MULTI-AGENT CATEGORY GAME MODEL

We consider a population of 100 agents, each of whom has an interpretation of 11 basic labels in a conceptual space. We take and d to be the Euclidean norm. This choice of metric space was originally inspired by other work done on colour spaces such as the RGB cube. Here, while we are not focussing on a specific application, considering a higher-dimensional space allows for a more general investigation of the proposed framework than, for example, the real line [1].

Each agent has an associated weight w in the range $[0.1, 0.9]$ quantifying the level of importance other agents give to their assertions (we do not use the range to prevent singularities occurring in the updating algorithms with division by 0). The simulations progress in discrete time steps, and at each time step every agent interacts with another in a pairwise manner in a language game (so 50 interactions take place at each time step), where interactions are determined at random modelling the population on a fully connected network. The weights of the agents are distributed uniformly in the range $[0.1, 0.9]$ at time step 0, and at each time step the weight of each agent increases by $0.8T^{-1}$ where T is the number of time steps in the simulation. If an agent's weight surpasses the maximal value of 0.9 then it is reset to 0.1.

3.1 Assertion and Updating Algorithms

For each interaction the two agents are presented with the same element x sampled from a uniform density on Ω. The speaker then asserts "x is θ" according to an assertion model based on their current interpretation I of the labels in LA. This listener then calculates an appropriateness measure $\mu_\theta x$, a subjective belief that the expression θ is appropriate to describe x, according to their interpretation I. If $\mu_\theta(x) < w$ where w is the weight of the speaking agent, then the listener updates their interpretation of LA so that the inequality $\mu_\theta(x) \geq w$ holds. The set of possible assertion expressions is made up of "x is L_i" (positive assertion) where L_i is a label in LA, and "x is $\neg L_i$" (negative assertion) where L_i is a label in LA. We define probability p to be prior probability of an agent asserting a positive assertion, and the probability $1 - p$ of being the probability of asserting a negative assertion. A speaking agent

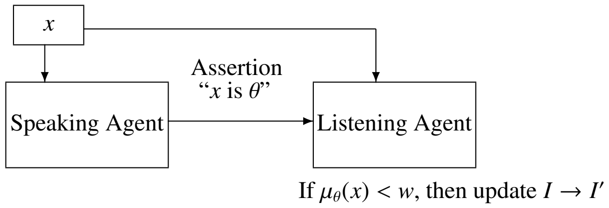

Figure 1. The Category Game Model

asserts an expression for x which gives maximal probability over a probability density on assertion expressions.

If an update is performed given an assertion of a basic label, then the listening agent must update their interpretation of LA so that $\mu_{L_i}(X) \geq w$ holds. In this case the agent moves the prototype P_i along the vector $\overrightarrow{P_i x}$ to a new prototype P_i' and increases the boundary variable ϵ_i to $\alpha\epsilon_i$, where $\alpha \geq 1$.

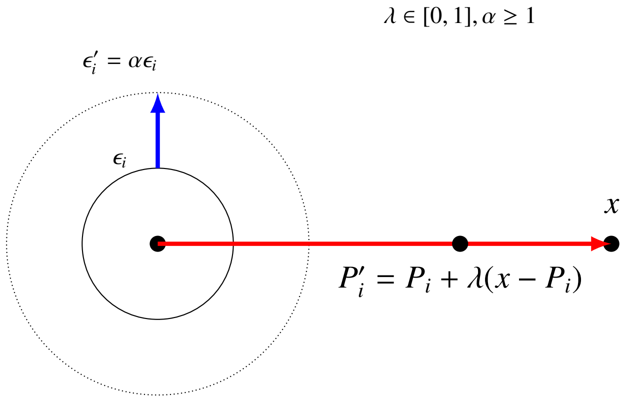
$\lambda \in [0, 1], \alpha \geq 1$
$\epsilon_i' = \alpha\epsilon_i$
ϵ_i
x
$P_i' = P_i + \lambda(x - P_i)$

Figure 2. A Positive Updating Rule

If a negative assertion is given to the listening agent, then an update must satisfy the constraint $\mu_{\neg L_i}(x) \geq w$ or $\mu_{L_i}(x) < w$. In this case the agent moves the prototype along the vector $-\overrightarrow{P_i x}$ to a new prototype P_i', and decreases the boundary variable ϵ_i to $\alpha\epsilon_i$ where $\alpha \leq 1$. Note that the convexity of the category neighbourhoods is retained through these updating rules.

3.2 Measuring Communication Success

To be a true measure of population agreement a measure of communication success should remain independent from the learning process taking place at each time step through the category game. So at each time step after the set of pairwise category games has taken place, a new set of 50 agent pairs is selected at random from the population. Each pair of agents is presented with 10 objects selected uniformly from Ω. Both agents select the label $\arg\max_{L_i}\{\mu_{L_i}(x) : i = 1, ..., n\}$, the label with maximal appropriateness measure, for each object presented to the agents. The proportion of objects for which the agents agree on this label gives a measure of agreement in interpretation of *LA* for these two agents. An average of this measure over all agent pairs gives our measure of *communication success*.

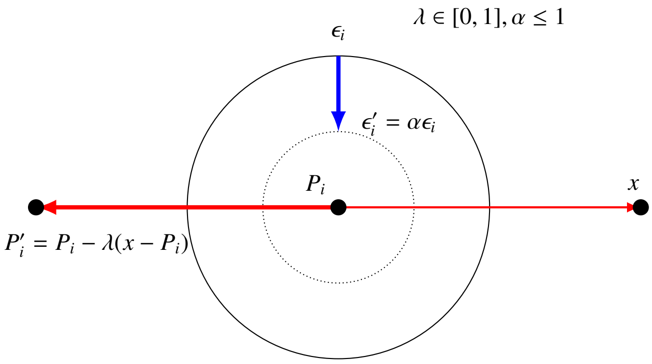
$\lambda \in [0, 1], \alpha \leq 1$
ϵ_i
$\epsilon_i' = \alpha\epsilon_i$
P_i
x
$P_i' = P_i - \lambda(x - P_i)$

Figure 3. A Negative Updating Rule

4 SIMULATION RESULTS

We ran simulations on a population of 100 agents for 10000 time steps. Different assertion priors were tested and each result is averaged over 25 runs. At time step 0 agents are given an interpretation of 11 labels with prototypes distributed according to a uniform distribution on Ω, and boundaries ϵ_i are each modelled by a uniform distribution on $[0, b]$ where b is given at time step 0 by a random number between 0 and 2.5. Figure 4 shows population communication success across time for $p \in \{0, 0.25, 0.5, 0.75, 1\}$. Initially the population shows average communication success of just below 0.1 which is to be expected as this corresponds to agents choosing a label at random for any presented object. We see an increase in communication success for all priors tested in figure 4, but the increase is greatest when $p = 0.75$ and $p = 1$. The plots here

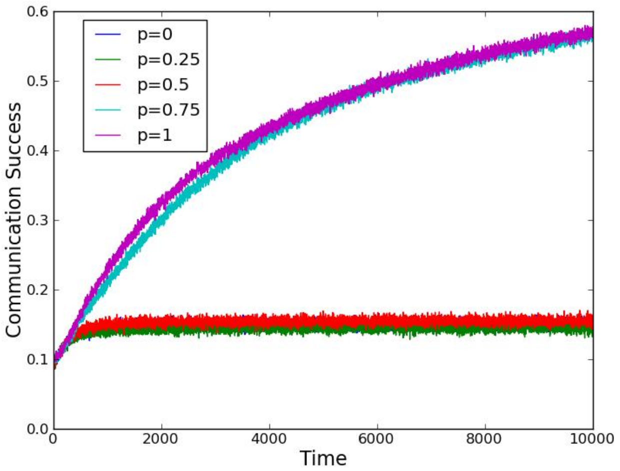

Figure 4. Communication Success over time with $p \in \{0, 0.25, 0.5, 0.75, 1\}$

do not suggest continuous behaviour of our performance metric with p.

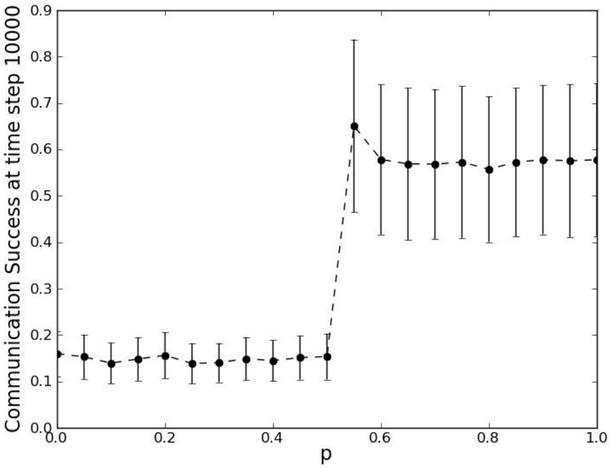

Figure 5. Communication Success at time step 10000 for priors $p \in [0, 1]$

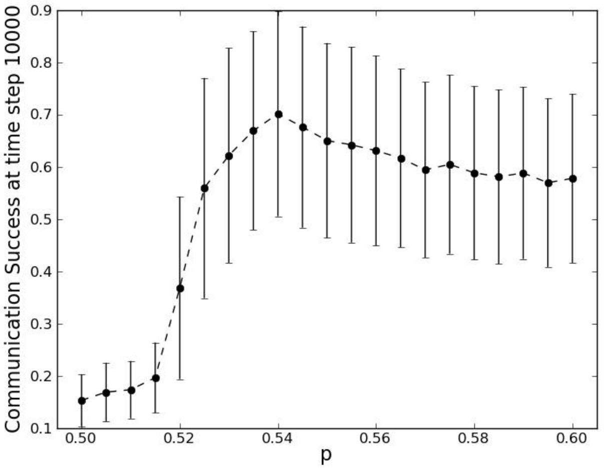

Figure 6. Communication Success at time step 10000 for priors $p \in [0.5, 0.6]$

As we see an increase in communication success for all priors we may consider the value of communication success at time step 10000 as a measure of performance for a simulation. Figure 5 shows this result for priors in increments of 0.05. We see that the best result is observed for $p = 0.55$. It is still not clear that there is continuity in out metric as a function of p, and so we focus on the range $p \in [0.5, 0.6]$. We see in figure 6 that for this range the highest value of communication success is seen when $p = 0.54$.

Figure 7 shows the average communication success across time when $p = 0$, when we see the smallest increase in communciation success. The error bars suggest that the increase is not greatly significant across time. Figure 8 shows the average communication success across time when $p = 0.54$, the best result recorded at the end of simulations. Error bars suggest that the average communication success is in this case significantly different at the end of simulations to the beginning of simulations.

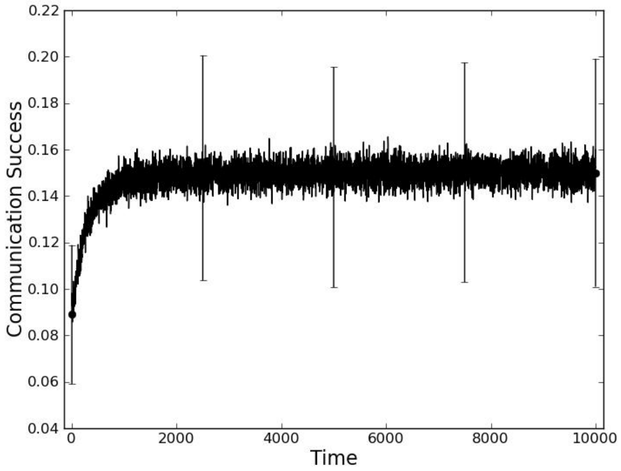

Figure 7. Communication Success over time with $p = 0$

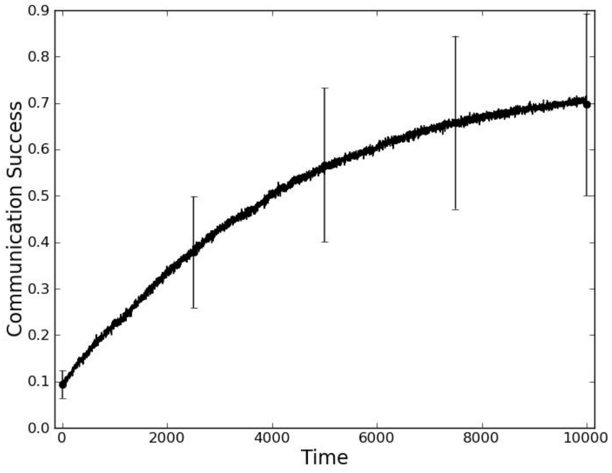

Figure 8. Communication Success over time with $p = 0.54$

5 CONCLUSION

Using a neighbourhood category representation we allow agents to model categories which allow for semantic vagueness. The range of language expressions generated using the label semantics framework here provide a richer description of the conceptual space.

Experimental results show that the average communication success in the population reaches the highest value when a mixed prior is used, allowing agents to assert both basic labels and their negations. This suggests that full use of these richer expressions can be the best choice of an agent to describe a simulated environment rather than a set of basic descriptions. Considering the communication success across time for the population, we see a significant difference in the metric at the beginning and end of simulations using a mixed asssertion prior. This suggests that the proposed framework is effective in allowing agents to develop a shared interpretation, allowing for successful communication in a population.

ACKNOWLEDGEMENTS

This work was supported by the EPSRC grant number EP/E501214/1.

REFERENCES

[1] A Baronchelli, T Gong, A Puglisi, and V Loreto, 'Modeling the emergence of universality in color naming patterns', *Proceedings of the National Academy of Sciences of the United States of America*, **107**(6), 2403–2407, (2010).

[2] T Belpaeme and J. Bleys, 'Explaining Universal Color Categories Through a Constrained Acquisition Process', *Adaptive Behavior*, **13**(4), 293–310, (2005).

[3] G. Berlin and P. Kay, *Basic Color Terms: Their Universality and Evolution*, University of California Press, 1969.

[4] R. S. Cook, P. Kay, and T. Regier, 'The World Color Survey Database', in *Handbook of Categorization in Cognitive Science*, eds., Henri Cohen and Claire Lefebvre, 223–241, Elsevier Science Ltd, (2005).

[5] I. Douven, L. Decock, R. Dietz, and P. gr, 'Vagueness: A conceptual spaces approach', *Journal of Philosophical Logic*, 1–24. 10.1007/s10992-011-9216-0.

[6] M. Dowman, 'Investigating the Effect of Random Noise on the Evolution of Colour Terms', in *Proceedings of IEEE Congress on Evolutionary Computation*, (2005).

[7] P. Gärdenfors, *Conceptual Spaces: The Geometry of Thought*, volume 106, MIT Press, 2000.

[8] J. Hampton, 'Prototype models of concept representation', in *Categories and Concepts Theoretical views and inductive data analysis*, eds., Van Mechelen, J A Hampton, and P Theuns R S Michlanski, 68–95, Academic Press, (1993).

[9] G. Lakoff, *Women, fire, and dangerous things*, University of Chicago Press, 1987.

[10] J. Lawry, 'An Overview of Computing with Words using Label Semantics', *Studies in Fuzziness and Soft Computing*, **220/2008**, 65–87, (2008).

[11] J. Lawry and Y. Tang, 'Uncertainty modelling for vague concepts: A prototype theory approach', *Artificial Intelligence*, **173**(18), 1539–1558, (2009).

[12] D. N. Osherson and E. E. Smith, 'On the adequacy of prototype theory as a theory of concepts', *Cognition*, **9**(1), 35–58, (1981).

[13] A. Puglisi, A. Baronchelli, and V. Loreto, 'Cultural route to the emergence of linguistic categories', *Proceedings of the National Academy of Sciences of the United States of America*, **105**(23), 7936–7940, (2008).

[14] E. Rosch Heider and D. C. Olivier, 'The structure of the color space in naming and memory for two languages', *Cognitive Psychology*, **3**(2), 337–354, (1972).

[15] L. Steels, 'A self-organizing spatial vocabulary', *Artificial Life*, **2**(3), 319–332, (1995).

[16] L. Steels, *Experiments in Cultural Language Evolution*, Advances in Interaction Studies, John Benjamins Publishing Company, 2012.

[17] L. Steels and T. Belpaeme, 'Coordinating perceptually grounded categories through language: a case study for colour.', *The Behavioral and brain sciences*, **28**(4), 469–89; discussion 489–529, (August 2005).

[18] K. Van Deemter, 'Utility and Language Generation: The Case of Vagueness', *Journal of Philosophical Logic*, **38**(6), 607–632, (2009).

[19] L. Wittgenstein, *Philosophical Investigations*, Macmillan, New York, 1953.

The Utility of Hedged Assertions in the Emergence of Shared Categorical Labels

Martha Lewis and **Jonathan Lawry** [1]

Abstract. We investigate the emergence of shared concepts in a community of language users using a multi-agent simulation. We extend results showing that negated assertions are of use in developing shared categories, to include assertions modified by linguistic hedges. Results show that using hedged assertions positively affects the emergence of shared categories in two distinct ways. Firstly, using contraction hedges like ‘very’ gives better convergence over time. Secondly, using expansion hedges such as ‘quite’ reduces concept overlap. However, both these improvements come at a cost of slower speed of development.

1 INTRODUCTION

An evolutionary approach to semantics enables the development in robots and autonomous agents of flexible, mutable concepts that could be learnt through interaction and can change over time [12]. This approach is investigated by Eyre and Lawry in [2], in which they develop a model of language evolution based in the label semantics framework. They show that using a mixture of positive and negated assertions enables the development of languages that are both shared, and discriminate effectively between elements within the environment. We extend this work to include assertions modified by the words ‘very’ and ‘quite’, and show that doing so improves performance in two ways. Use of the hedge ‘very’ improves levels of convergence attained. Using the hedge ‘quite’ reduces the amount of overlap within an agent’s label set. We describe in detail the theoretical approach to concepts taken and linguistic hedges in the remainder of this section. Section 2 gives details of the mathematical and computational model used in the simulations. Section 3 gives results of the simulations which are discussed in section 4. Lastly, section 5 gives conclusions and further avenues of research.

1.1 A representation model for concepts

We model concepts within the label semantics framework [7, 8], combined with prototype theory [10] and the conceptual spaces model of concepts [3]. Prototype theory offers an alternative to the classical theory of concepts, basing categorization on proximity to a prototype. This approach is based on experimental results where human subjects were found to view membership in a concept as a matter of degree, with some objects having higher membership than others [10]. Fuzzy set theory [14], in which an object x has a graded membership $\mu_L(x)$ in a concept L, was proposed as a formalism for prototype theory. However, numerous objections to its suitability have been made [9, 11, 6, 4, 5].

Conceptual spaces theory renders concepts as convex regions of a *conceptual space* - a geometrical structure with quality dimensions and a distance metric. Examples are: the RGB colour cube, pictured in figure 1; physical dimensions of height, breadth and depth; or the taste tetrahedron. Since concepts are convex regions of such spaces, the centroid of such a region can naturally be viewed as the prototype of the concept.

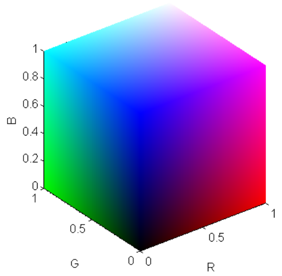

Figure 1. The RGB cube represents colours in three dimensions of Red, Green and Blue. A colour concept such as ‘purple’ can be represented as a region of this conceptual space.

Label semantics [7] is a random set approach to concepts which quantifies an agent’s uncertainty about the extent of application of a concept. We refer to this as *subjective uncertainty* [8] to emphasise that it concerns the definition of concepts and categories, in contrast to stochastic uncertainty which concerns the state of the world. Lawry and Tang [8] combine the label semantics approach with conceptual spaces and prototype theory, to give a formalisation of concepts as based on a prototype and a threshold, located in a conceptual space.

Within this framework, agents use sets of labels $LA = \{L_1, L_2, ..., L_n\}$ to describe an underlying conceptual space Ω with distance metric $d(x, y)$ between points. The conceptual space could be, as mentioned, the RGB colour space. Labels L_i would then be concepts such as ‘red’, ‘blue’, ‘purple’, ‘orange’ and so on. These labels are viewed as regions of the conceptual space. So the concept ‘blue’ is represented by the blue region in the colour cube. Within label semantics, these regions are specified by prototypes P_i and thresholds ε_i. This is in contrast to Gärdenfors’ original approach which is to view the space as partitioned by a Voronoi tessellation. If this latter approach is taken, each individual point in the conceptual space is allocated to exactly one label. With a prototype-threshold approach, it is easy to accommodate the idea of an object being accurately described by more than one concept, or conversely, some

---

[1] University of Bristol, England, email: martha.lewis@bristol.ac.uk, j.lawry@bristol.ac.uk

points within the space not being assigned to any concept. This difference is illustrated in figures 2 and 3.

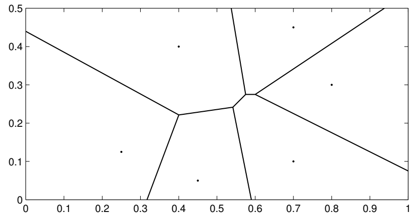

Figure 2. Conceptual space divided into concepts according to a Voronoi tessellation around prototypes. Each part of the space corresponds to exactly one concept.

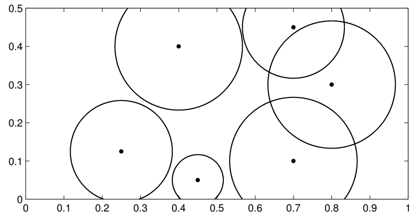

Figure 3. Conceptual space divided into concepts according to a prototype-threshold approach. Some points in the space correspond to more than one concept, and some correspond to none.

In this model, however, agents are uncertain as to exactly where the thresholds lie. To illustrate this, consider the concept ‘tall’. It is easy to point out a tall person, and to point out a person who is not tall, but it is difficult to specify the exact threshold between ‘tall’ and ‘not tall’. This uncertainty concerning where the threshold lies is represented in the label semantics framework by saying that a threshold ε_i is drawn from a probability distribution δ_i. Labels L_i are associated with neighbourhoods $\mathcal{N}_{L_i}^{\varepsilon_i} = \{\vec{x} \in \Omega : d(\vec{x}, P_i) \leq \varepsilon_i\}$, i.e. the region within the threshold. These ideas are represented in figure 4.

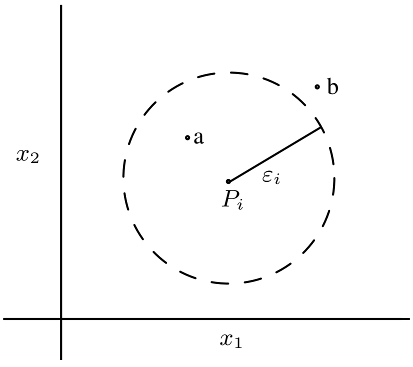

Figure 4. Prototype-threshold representation of a concept L_i. The conceptual space has dimensions x_1 and x_2. The concept has prototype P_i and threshold ε_i. The uncertainty about the threshold is represented by the dotted line. The neighbourhood $\mathcal{N}_{L_i}^{\varepsilon_i}$ is the area within the dotted line. Element a in the conceptual space is within the threshold, so it is appropriate to assert ‘a is L_i’. Element b is outside the threshold, so it is not appropriate to assert ‘b is L_i’

The threshold ε_i is uncertain, however, so there is some probability that ε_i in figure 4 is actually wide enough to include the object b, i.e. that L_i is appropriate to describe b. The appropriateness $\mu_{L_i}(x)$ of a label L_i to describe an element x is then given by the probability that x lies within the neighbourhood $\mathcal{N}_{L_i}^{\varepsilon_i}$, i.e. that the distance $d(x, P_i)$ is less than ε_i. So:

$$\mu_{L_i}(x) = P(d(x, P_i) \leq \varepsilon_i) = \int_{d(x,P_i)}^{\infty} \delta_i(\varepsilon_i)\mathrm{d}\varepsilon_i$$

Figure 5 shows how this appropriateness measure works in a setup similar to that in figure 4.

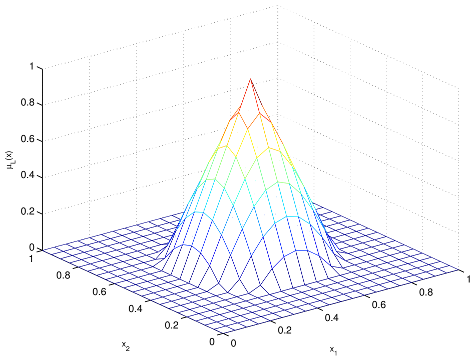

Figure 5. Membership in a concept. The prototype of the label is at $[0.5, 0.5]$ and the threshold ε has distribution $U[0, 0.3]$. When $x = [0.5, 0.5]$, $\mu_L(x) = 1$. As we move further away from the prototype, membership in the concept decreases, and is 0 when $d(x, P_i) > 0.3$

This appropriateness measure is similar to Zadeh’s description of fuzzy membership in a concept [14].

1.2 Linguistic hedges

Hedges are words or phrases such as ‘very’, ‘quite’, ‘strictly speaking’ which modify the domain of application of a concept. In particular, ‘very’, and ‘quite’ respectively contract or expand the domain of application of a concept, so that, for example, ‘very tall’ applies to fewer people than does ‘tall’, whereas ‘quite tall’ applies to more. Within fuzzy set theory, we expect that $\mu_{\text{very } L}(x) \leq \mu_L(x)$ and $\mu_{\text{quite } L}(x) \geq \mu_L(x)$. Applying this to the concept ‘tall’, again, this means that membership in the concept ‘very tall’ should always be less than membership in ‘tall’. So anyone who can be described as ‘very tall’ can also be described as ‘tall’. Zadeh [15] uses operations of *concentration* and *dilation* to render these ideas. Concentration is described as $CON(\mu_{L_i}(x)) = (\mu_{L_i}(x))^2$ and dilation is often rendered as $DIL(\mu_{L_i}(x)) = (\mu_{L_i}(x))^{1/2}$. However, we argue, as do [1], that Zadeh’s formulae are, to an extent, arbitrary, since the notion of taking a power of a membership value does not correspond to anything that language users might do. Rather, it simply has some of the right effects. As with [1], we take a semantic approach.

We propose that a concept ‘very L’ or ‘quite L’ be rendered by considering that the prototype of ‘very/quite L’ is equal to that of the base concept L, but that the threshold of the hedged concept ‘very/quite L’ is respectively smaller or larger than that of the base concept. This approach is grounded in the idea that ‘very/quite L’ should apply to respectively fewer or more objects than L. Narrowing or widening the threshold achieves this in a natural way. This is illustrated in figure 6.

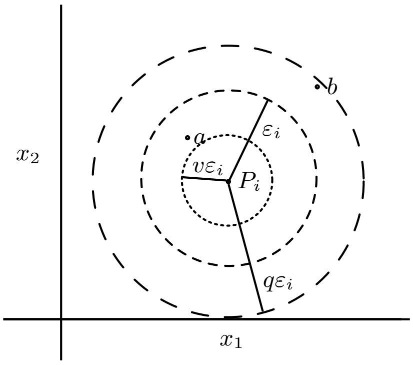

Figure 6. Representation of 'very L_i' and 'quite L_i'. 'Very L_i' has prototype P_i and threshold $v\varepsilon_i \leq \varepsilon_i$. 'Quite L_i' has prototype P_i and threshold $q\varepsilon_i \geq \varepsilon_i$. Notice that although L_i is appropriate to describe a, vL_i is not. Also, although L_i is not appropriate to describe b, qL_i is.

Our model of the hedges 'very' and 'quite' therefore requires simply that $v\varepsilon_i \leq \varepsilon_i$ and that $q\varepsilon_i \geq \varepsilon_i$. We implement this model in a version of the multi-agent simulation created in [2] in order to investigate how the use of these hedges in a model of language helps the emergence of shared categories across a community of language users.

2 METHODS

2.1 Overview

To investigate the utility of hedged assertions we implement a multi-agent simulation of a version of the category game [13], following [2], in which shared categories develop over time as a result of the interactions of the category users. An overview of the game is as follows. Agents use labels to describe a conceptual space Ω. At each timestep, agents are randomly paired into speakers and listeners, and each pair is shown a distinct element $x \in \Omega$. The speaker makes an assertion θ about the element based on its label set. The listener then updates its own label set to be more similar to that of the speaker, based on this assertion and a parameter w which can be thought of as the age of the speaker. The update made by the speaker is a combination of shifting the prototype of the relevant label and changing the size of the threshold. The aim is that after a number of timesteps, label sets across the population have converged to a common set of shared categories.

2.2 Conceptual models

Each agent is equipped with the same number n of labels L_i, with point prototypes $P_i \in \Omega$, where $\Omega = [0, 1]^3$. At the start of the simulations the P_i are uniformly distributed around the space. Thresholds ε_i are also randomly initiated, and considered to be some multiple of a base threshold ε. Each threshold $\varepsilon_i \sim U(0, b_i)$, where again, the b_i can be considered to be a multiple of some common b, and the b_i are taken from $U[0.5, 2]$. The distance metric is Euclidean.

Each agent therefore has a label set $LA = \{L_1, L_2, ...L_n\}$. These labels can be hedged to form a set $LA^+ = LA \cup \{\text{very } L_i, \text{quite } L_i : i = 1, ..., n\}$. Hedged concepts have the same prototype P_i as basic labels, but a scaled threshold $v\varepsilon_i$ or $q\varepsilon_i$ where $v < 1$ and $q > 1$. Agents can assert positive or negated, hedged or basic labels, giving an assertion set $AS = \{kL_i, \neg kL_i : i = 1, ..., n; k = \text{very, quite, basic}\}$, where $k = \text{basic}$ means that the label is not hedged.

2.3 Assertion model

At each timestep, half the agents are designated speaker agents and make assertions, determined by the assertion model used.The assertion model is based on the probability of making a particular assertion θ, given that the object being described is x. Following methods in [2, 8], we calculate the posterior probability of each $\theta \in AS$, given an element $x \in \Omega$. The assertion made by a speaker agent is the assertion with the highest probability. The posterior probability of each θ, given x, is determined by the appropriateness of the assertion θ to describe x, i.e. $\mu_\theta(x)$, and the prior probability $P(\theta)$ of asserting θ.

We first consider which sets of labels that are appropriate to describe $x \in \Omega$. The probability that any particular set of labels $F \subseteq LA$ are appropriate to describe x is given by a probability mass function $m_x : 2^{LA} \to [0, 1]$. One way of determining m_x is via the *consonant selection function* introduced in [8]. This states:

Definition 1 (Consonant selection function) *Given non-zero appropriateness measures on basic labels* $\mu_{L_i}(x) : i = 1, ..., n$ *ordered such that* $\mu_{L_i}(x) \geq \mu_{L_{i+1}}$ *for* $i = 1, ..., n$*, the consonant selection function identifies the mass function*

$$m_x(\{L_1, ..., L_n\}) = \mu_{L_n}(x)$$

$$m_x(\{L_1, ..., L_i\}) = \mu_{L_i}(x) - \mu_{L_{i+1}}(x) \textit{ for } i = 1, ..n - 1$$

$$m_x(\emptyset) = 1 - \mu_{L_1}(x)$$

$$m_x(F) = 0 \textit{ if } F \neq \{L_1, L_2, ..., L_k\} \textit{ for some } k \leq n$$

Because we have ordered the labels by $\mu_{L_i}(x) \geq \mu_{L_{i+1}}$, if the label L_i is appropriate to describe x, all labels $L_j : j < i$ must also be appropriate to describe x. The quantity $\mu_{L_i}(x) - \mu_{L_{i+1}}(x)$ corresponds to the idea that x in some sense lies between the thresholds ε_{i+1} and ε_i, so that L_i is appropriate to describe x, but L_{i+1} is not. We extend this definition to the case of hedged labels simply by considering all hedged labels as basic labels, explained in the example below.

Example 2 (Determining the mass function) *Suppose we are determining the mass function for subsets* $F \subseteq \{kL_1, kL_2 : k = \textit{very}, \textit{quite}, \textit{basic}\}$*, given the point* $a \in x_1 \times x_2$*, as illustrated in figure 7.*

Suppose that $\mu_{\textit{quite } L_2}(a) = 0.9$*,* $\mu_{L_2}(a) = 0.7$*,* $\mu_{\textit{quite } L_1}(a) = 0.3$*,* $\mu_{\textit{very } L_2}(a) = 0.1$*,* $\mu_{L_1}(a) = 0$*,* $\mu_{\textit{very } L_1}(a) = 0$*, giving us the order* $\mu_{\textit{quite } L_2}(a) \geq \mu_{L_2}(a) \geq \mu_{\textit{quite } L_1}(a) \geq \mu_{\textit{very } L_2}(a) \geq \mu_{L_1}(a) \geq \mu_{\textit{very } L_1}(a)$*. We may then assign probabilities to subsets of labels according to the consonant selection function:*

$$\begin{aligned} m_x(F_6) &= m_x(\{\textit{quite } L_2, L_2, \textit{quite } L_1, \textit{very } L_2, L_1, \textit{very } L_1\}) \\ &= \mu_{\textit{very } L_1}(a) = 0 \\ m_x(F_5) &= m_x(\{\textit{quite } L_2, L_2, \textit{quite } L_1, \textit{very } L_2, L_1\}) \\ &= \mu L_1(a) - \mu_{\textit{very } L_1}(a) = 0 \\ m_x(F_4) &= m_x(\{\textit{quite } L_2, L_2, \textit{quite } L_1, \textit{very } L_2\}) \\ &= \mu_{\textit{very } L_2}(a) - \mu_{L_1}(a) = 0.1 \\ m_x(F_3) &= m_x(\{\textit{quite } L_2, L_2, \textit{quite } L_1\}) \\ &= \mu_{\textit{quite } L_1}(a) - \mu_{\textit{very } L_2}(a) = 0.2 \\ m_x(F_2) &= m_x(\{\textit{quite } L_2, L_2\}) = \mu_{L_2}(a) - \mu_{\textit{quite } L_1}(a) = 0.4 \\ m_x(F_1) &= m_x(\{\textit{quite } L_2\}) = \mu_{\textit{quite } L_2}(a) - \mu_{L_2}(a) = 0.2 \\ m_x(\emptyset) &= 1 - \mu_{\textit{quite } L_2}(a) = 0.1 \end{aligned}$$

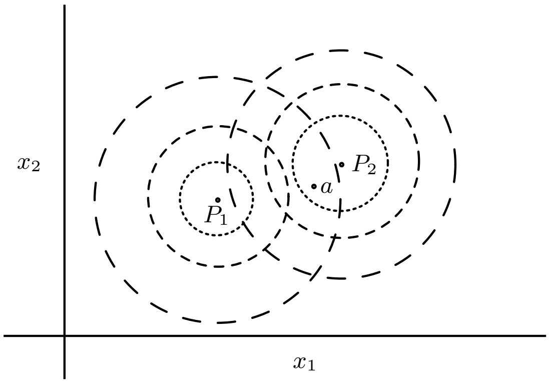

Figure 7. Determining the mass function on subsets of $\{kL_1, kL_2 : k = \text{very, quite, basic}\}$. P_1 and P_2 represent prototypes for each lable L_1 and L_2, and the dotted lines give the different thresholds according to the hedges, as in figure 6. Notice that 'quite L_2', L_2, 'very L_2' and 'quite L_1' are all appropriate to describe a, although with different appropriateness measures (not shown), but L_1 and 'very L_1' are not.

Having determined the probability mass function on sets of labels, a mass assignment on sets of assertions is then defined.

Definition 3 (Mass assignment on assertions) $ma_x : 2^{AS} \rightarrow [0,1]$ *is defined such that:*

$$ma_x(G) = \sum_{F \subseteq LA^+ : \mathscr{C}(F) = G} m_x(F)$$

where $\mathscr{C}(F) = \{\theta \in AS : F \in \lambda(\theta)\}$, *and* $\lambda(\theta)$ *is defined recursively by*

$$\begin{aligned} \lambda(kL_i) &= \{F \subseteq LA^+ : kLi \in F\} \\ \lambda(\neg\theta) &= (\lambda(\theta))^c \\ \lambda(\theta \wedge \phi) &= \lambda(\theta) \cap \lambda(\phi) \\ \lambda(\theta \vee \phi) &= \lambda(\theta) \cup \lambda(\phi) \end{aligned}$$

This definition has the implication that for $G_i = F_i \cup \{\neg kL_j : kL_j \in LA^+ \backslash F_i\}$, $ma_x(G_i) = m_x(F_i)$.

Then the probability of an assertion θ being made, given that an object x is being described, can be calculated by summing over $G \subseteq AS$ that contain θ.

Definition 4 *Given a prior distribution on AS, a posterior distribution given an object x can be calculated by:*

$$\begin{aligned} P(\mathscr{A} = \theta | x) &= \sum_{G \subseteq AS : \theta \in G} ma_x(G) P(\mathscr{A} = \theta | \mathscr{A} \in G) \\ &= P(\theta) \sum_{G \subseteq AS : \theta \in G} \frac{ma_x(G)}{P(G)} \end{aligned}$$

Here, $P(G) = \sum_{\varphi \in G} P(\varphi)$

The value of $P(\theta)$ for one particular label L is a product of two elements: the prior probability pp of making a positive assertion (or $1 - pp$ for a negated assertion); and the prior probability of making a hedged assertion, given by pv for making an assertion hedged with the word 'very', pq for making an assertion hedged with 'quite' or $1 - pv - pq$ for making a basic assertion, summarised in table 1.

Table 1. Prior probabilities of each type of assertion $\pm kLi$

$*$	pv	pb	pq
pp	$P(vL)$	$P(L)$	$P(qL)$
pn	$P(\neg vL)$	$P(\neg L)$	$P(\neg qL)$

The prior probability of asserting any particular label $L_i \in LA$ is uniform across LA. Hence the value of $P(\theta)$ calculated above should be divided by n, giving, for example,

$$P(\neg vL_2) = \frac{pn * pv}{n}$$

Example 5 (Determining the posterior probability of assertion) *Suppose, for an easy example, we want to calculate the probability of asserting 'very L_1', given object a, as in example 2. We need to calculate*

$$P(\mathscr{A} = \text{'very } L_1\text{'} | a) = P(\theta) \sum_{G \subseteq AS : \text{'very } L_1\text{'} \in G} \frac{ma_x(G)}{P(G)}$$

where $G_i = F_i \cup \{\neg kL_j : kL_j \in LA^+ \backslash F_i\}$. *However, the only subset* $G_i \ni$ *'very L_1' is* G_6, *so*

$$\begin{aligned} P(\mathscr{A} = \text{'very } L_1\text{'} | x) &= P(\text{'very } L_1\text{'}) \frac{ma_x(G_6)}{P(G_6)} \\ &= 0 \end{aligned}$$

Suppose, for a more involved example, the label set LA^+ is as in example 2, with $pp = 0.7$, $pv = 0.7$, $pq = 0.2$, *and we want to determine* $P(\mathscr{A} = \text{quite } L_1 | a)$. *The prior probability* $P(\text{quite } L_1) = \frac{0.7 * 0.2}{2} = 0.07$. *So we have:*

$$\begin{aligned} &P(\mathscr{A} = \text{quite } L_1 | a) \\ &= 0.07 \sum_{G_i : \text{quite } L_1 \in G_i} \frac{ma_x(G_i)}{P(G_i)} \\ &= 0.07(\frac{ma_x(G_6)}{P(G_6)} + \frac{ma_x(G_5)}{P(G_5)} + \frac{ma_x(G_4)}{P(G_4)} + \frac{ma_x(G_3)}{P(G_3)}) \\ &= 0.07(0 + 0 + \frac{0.1}{\sum_{\varphi \in G_4} P(\varphi)} + \frac{0.2}{\sum_{\varphi \in G_3} P(\varphi)}) \\ &= 0.07(\frac{0.1}{0.54} + \frac{0.2}{0.4}) \\ &= 0.048 \end{aligned}$$

Having calculated the probability of each assertion, the speaker agent makes the most probable assertion $\theta \in AS$.

2.4 Updating algorithms

Once the speaker agent has made assertion θ, the listener agent computes $\mu_\theta(x)$ based on its current label set. If $\mu_\theta(x) < w$, where w is a parameter that can be thought of as the age of the speaker agent, the listener agent updates its label set LA by moving the prototype and/or changing the threshold of the concept, until $\mu_\theta(x) = w$. Formulae for these updates are again based on [2]. A label defined by P_i and ε_i is updated to $P_i' = P_i - \lambda(x - P_i)$ and $\varepsilon_i' = \alpha\varepsilon_i$. Values for λ and α are sought, such that $\mu_\theta'(x) = w$.

2.4.1 Case 1: $\theta = kL_i$

Recall that $\varepsilon_i \sim U(0, b_i)$, so that for $x \in \Omega$,

$$\mu_{kL_i}(x) = 1 - \frac{||x - P_i||}{kb_i} < w \text{ by assumption.}$$

The label L_i is updated to L_i', where $P_i' = P_i - \lambda(x - P_i)$ and $\varepsilon_i' = \alpha\varepsilon_i$, such that $\mu_{kL_i'}(x) \geq w$, and minimising the distance between the interpretations as measured by the Haussdorff distance between the two neighbourhoods,

$$\begin{aligned} \mathscr{H}(\mathcal{N}_{L_i}, \mathcal{N}_{L_i'}) &= ||P_i - P_i'|| + |\varepsilon_i - \varepsilon_i'| \qquad (1) \\ &= |\lambda|||x - P_i|| + \frac{\varepsilon b_i}{b}|1 - \alpha| \ (*) \end{aligned}$$

To minimise the update, we set $\mu_{kL_i'}(x) = w$, so:

$$w = \mu_{kL_i'}(x) = 1 - \frac{||x - P_i'||}{kb_i'} = 1 - \frac{|1 - \lambda|||x - P_i||}{\alpha k b_i}$$

which gives

$$\begin{aligned} \alpha &= \frac{|1 - \lambda|||x - P_i||}{(1 - w)kb_i} \\ &= \frac{(1 - \lambda)||x - P_i||}{(1 - w)kb_i} \quad \text{since } \lambda = 1 \to P_i' = x \end{aligned}$$

To update L_i we will always want $\lambda \geq 0, \alpha \geq 1$, as we are dealing with a positive label.

Substituting α into equation (*), we obtain

$$\begin{aligned} \mathscr{H}(\mathcal{N}_{L_i}, \mathcal{N}_{L_i}') &= |\lambda|||x - P_i|| + \frac{\varepsilon b_i}{b}\left(\frac{(1 - |\lambda|)||x - P_i||}{(1 - w)kb_i} - 1\right) \\ &= |\lambda|||x - P_i||\left(1 - \frac{\varepsilon}{b(1 - w)k}\right) + \frac{\varepsilon||x - P_i||}{b(1 - w)k} - \frac{\varepsilon b_i}{b} \qquad (2) \end{aligned}$$

Then if $1 - \frac{\varepsilon}{b(1-w)k} > 0$, i.e. $\varepsilon < b(1 - w)k$, the quantity (2) can be minimised by setting $\lambda = 0$ so $\alpha = \frac{||x-P_i||}{(1-w)kb_i}$. Otherwise, we have $\alpha = 1$, $\lambda = 1 - \frac{(1-w)kb_i}{||x-P_i||}$.

Since ε is a random variable, so is the choice between λ and α. We therefore need a concrete updating rule. We update P_i and b_i with the expected values of λ and α respectively. $\varepsilon \sim$ Uniform[0, b], so

$$P(\varepsilon < b(1 - w)k) = \begin{cases} (1 - w)k & \text{if } (1 - w)k < 1 \\ 1 & \text{otherwise} \end{cases}$$

We can therefore calculate

$$E(\alpha) = \begin{cases} \frac{||x-P_i||}{b_i} + 1 - (1 - w)k & \text{if } (1 - w)k < 1 \\ \frac{||x-P_i||}{(1-w)kb_i} & \text{otherwise} \end{cases}$$

and

$$E(\lambda) = \begin{cases} (1 - (1 - w)k)(1 - \frac{(1-w)kb_i}{||x-P_i||}) & \text{if } (1 - w)k < 1 \\ 0 & \text{otherwise} \end{cases}$$

2.4.2 Case 2: $\theta = \neg kL_i$

By an entirely similar argument, we obtain

$$E(\alpha) = \begin{cases} \frac{||x-P_i||}{b_i} + 1 - wk & \text{if } wk < 1 \\ \frac{||x-P_i||}{wkb_i} & \text{otherwise} \end{cases}$$

and

$$E(\lambda) = \begin{cases} (1 - wk)(1 - \frac{wkb_i}{||x-P_i||}) & \text{if } wq < 1 \\ 0 & \text{otherwise} \end{cases}$$

So at each timestep, each listener agent, for whom $\mu_\theta(x) < w$, updates the relevant label using the the quantities $E(\alpha)$, $E(\lambda)$.

2.5 Performance metrics

Performance metrics from [2] are used, measuring the Average Pairwise Distance between label sets (APD) and the Average Label Overlap (ALO). APD measures the difference in label sets in the community, and ALO indicates the extent to which an agent's concepts overlap. We seek low values for each metric.

APD is calculated using the Haussdorff distance between two neighbourhoods as given in equation 1. The difference between the label sets of any one pair of agents is given by

$$IPD = \sum_{i=1}^{n} \mathscr{H}(\mathcal{N}_{L_i}^{j}, \mathcal{N}_{L_i}^{k})$$

where n is the number of labels each agent has and j and k refer to distinct agents.

This is averaged over pairs of agents. There are N agents, therefore $\binom{N}{2}$ pairs, giving:

$$APD = \frac{2}{N(N - 1)} \sum_{k=j+1}^{N} \sum_{j=1}^{N} IPD_{jk}$$

ALO is the extent to which labels overlap. To calculate this, we take the maximum value of the intersection of a pair of labels, as measured by a min rule. We average this value over pairs of labels. The overlap within an individual's label set is therefore

$$ILO = \frac{2}{n(n - 1)} \sum_{j=i+1}^{n} \sum_{i=1}^{n} max\{min\{\mu_{L_i}(x), \mu_{L_j}(x) : x \in \Omega\}\}$$

Averaged across the population this is:

$$ALO = \frac{1}{N} \sum_{k=1}^{N} ILO_k$$

where ILO_k siginifies agent k's label overlap.

2.6 Simulation process

Simulations with $n = 100$ agents were run for $T = 10^4$ timesteps. Agent weights $w \in [0.2, 0.8]$ were updated at each timestep in increments of $1/T$. When $w \geq 0.8$, agents are reborn with randomised labels and $w = 0.2$. 20 simulations are run for each reported combination of parameters.

[2] show that if $pp \in [0.5, 0.6]$ then performance of the system changes from low ALO and high APD to vice versa at approximately $pp = 0.56$. We ran simulations in a slightly extended range for comparison, varying the prior probabilities pv, pb and pq of asserting the different hedges 'very', 'basic', and 'quite'. We present results from three sets of parameters. As a baseline we run simulations with no hedges, i.e. $pv = 0$, $pb = 1$, $pq = 0$. To investigate the effects of using contraction hedges, we run simulations with parameters $pv = 0.7$, $pb = 0.2$, $pq = 0.1$. For expansion hedges, we use parameters $pv = 0.1$, $pb = 0.2$, $pq = 0.7$.

3 RESULTS

The results presented show performance against the two metrics after 10^4 simulation timesteps. By this point, the population has generally reached a steady state in which performance does not greatly change.

Figure 8 shows the steady state of APD achieved after 10^4 timesteps for a range of values $pp \in [0.4, 0.6]$. Three sets of results are presented: results using unhedged assertions; results with a high prior probability of using contraction hedges; and results from simulations with a high prior probability of asserting expansion hedges, where these prior probabilities are as stated in 2.6.

A high prior of asserting contracted labels reduces minimum APD achieved from 0.38 when $pp = 0.56$ or $pp = 0.6$ to 0.29 when $pp = 0.57$ (figure 8). Performing a paired t-test across the 20 simulations gives the mean difference between these values as 0.097. This difference is statistically significant with $p < 0.001$ and with 95% confidence interval $[0.084, 0.110]$. The median and range of results are given in figure 9. At $pp = 0.57$, ALO decreases, from 0.92 to 0.89 (figure 11). The mean value of this difference across the 20 simulations is 0.032. Again, this is statistically significant with $p < 0.001$ and 95% confidence interval of $[0.029, 0.035]$, further illustrated in figure 10. These results imply that a high prior probability of asserting contraction hedges enables us to improve convergence between agents' label sets as well as reducing overlap within label sets slightly.

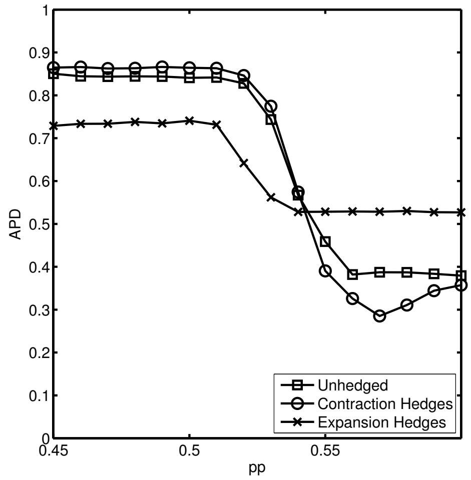

Figure 8. APD after 10^4 timesteps. Using contraction hedges reduces the minimum APD achieved from 0.38 at $pp = 0.56$ to 0.29 at $pp = 0.57$. Expansion hedges reduce APD from 0.85 to 0.73 at $pp = 0.45$

With a high prior probability of asserting expanded labels, lower values of ALO can be achieved when the probability of asserting positive labels is 0.45 , decreasing to 0.02 compared to 0.1, figure 11. The mean difference between these values across the 20 simulations is 0.083, which is statistically significant with $p < 0.001$ and a 95% confidence interval of $[0.078, 0.089]$. The data is represented in figure 13. At this value of pp, APD achieved is 0.73 compared to 0.85 for unhedged assertions, figure 8. The mean value of this difference across the 20 simulations is 0.12. This figure is statistically significant with $p < 0.001$ and 95% confidence interval $[0.116, 0.126]$. The data is again represented in figure 12. A high prior probability of asserting expansion hedges therefore enables minimal overlap to be maintained at low pp whilst improving convergence.

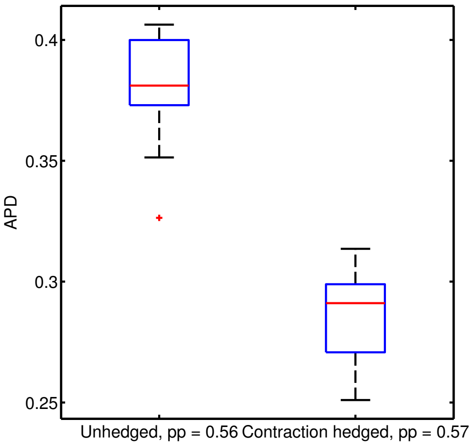

Figure 9. Using contraction hedges reduces minimum APD. Box and whisker plot of APD after 10^4 timesteps for 20 simulations, for $pp = 0.56$ unhedged, $pp = 0.57$ with a high probability of contraction hedges (values of pp at which minimum APD is achieved). The middle line shows median value, the box shows the 25th and 75th percentile. Whiskers show the range of data excluding outliers, and crosses show outliers.

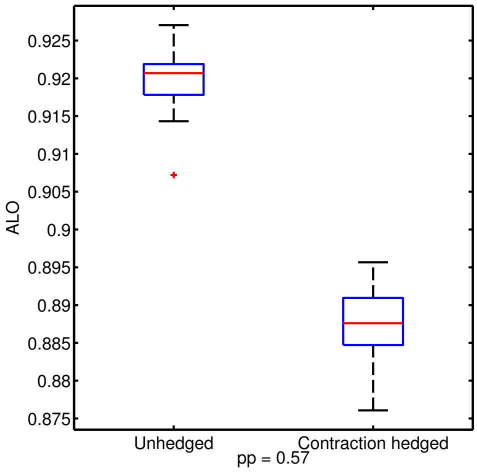

Figure 10. Using contraction hedges slightly reduces ALO. Box and whisker plot of ALO after 10^4 timesteps for 20 simulations, for $pp = 0.57$. The middle line shows median value, the box shows the 25th and 75th percentile. Whiskers show the range of data excluding outliers, and crosses show outliers.

We can also examine how fast the community of agents arrives at a steady state. Figure 14 shows that at short timescales ($t < 2000$), better convergence may be achieved allowing only unhedged assertions. In a more extreme case, figure 15 shows that for $pp = 0.5$, better performance on the ALO metric is only achieved after 7500 timesteps. Although this improvement takes a longer time to achieve, it goes together with improved performance on APD which is achieved in a similar timescale to the unhedged model 16.

4 DISCUSSION

These results show that, in a model of language development across a population, hedged assertions can improve both the level of convergence to shared language as measured by average pairwise dif-

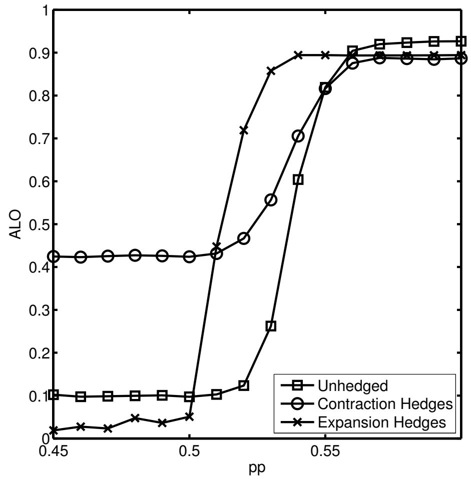

Figure 11. Contraction hedges slightly reduce high levels of ALO. At $pp = 0.57$, ALO is reduced from 0.92 to 0.89. Expansion hedges reduce minimum ALO from 0.1 for unhedged assertions to 0.02 for expansion hedged assertions, at $pp = 0.45$.

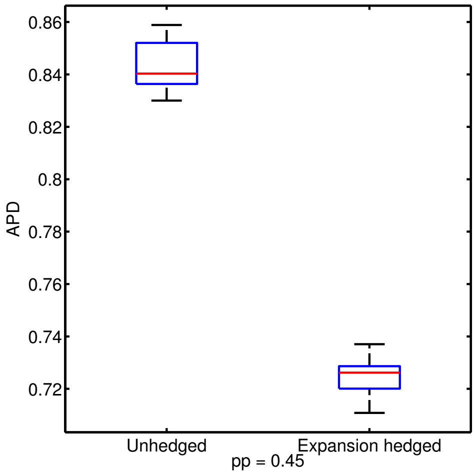

Figure 12. Expansion hedges reduce maximum APD. Box and whisker plot of APD after 10^4 timesteps for 20 simulations, for $pp = 0.45$. The middle line shows median value, the box shows the 25th and 75th percentile. Whiskers show the range of data excluding outliers, and crosses show outliers.

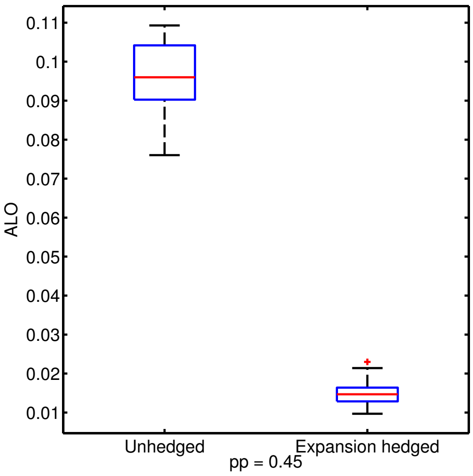

Figure 13. Expansion hedges reduce minimum ALO. Box and whisker plot of ALO after 10^4 timesteps for 20 simulations, for $pp = 0.45$. The middle line shows median value, the box shows the 25th and 75th percentile. Whiskers show the range of data excluding outliers, and crosses show outliers.

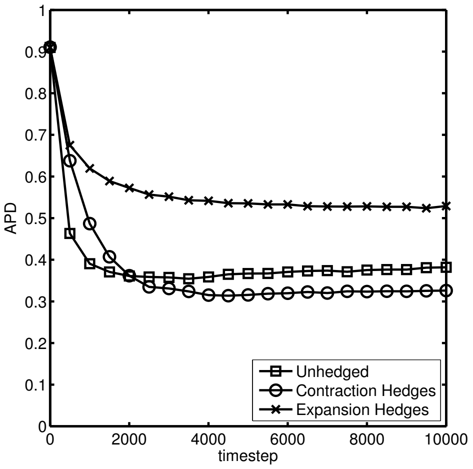

Figure 14. APD vs time for models with no hedged assertions, contracted assertions and expanded assertions, for a prior probability of positive assertions $pp = 0.56$. Although the final value reached is lower when there is a high probability of making contracted assertions, the community of agents takes longer to reach that value.

ference between label sets (APD) and, to an extent, the discriminatory power of individuals' label sets, as measured by average label overlap (ALO). The two different types of hedges improve performance in distinct ways. If overall convergence is important, a high prior probability of asserting contraction hedges should be used to improve performance on the APD metric. Conversely, if the ability of the agents to discriminate precisely between objects in the environment is more important, then expansion hedges, together with lower probabilities of asserting positive labels, should be used to maintain low levels of ALO whilst still improving performance on APD.

The improved performance against the two metrics is tempered by the fact that the speed at which the steady state is achieved is somewhat slower than when using simply unhedged assertions. However, the improvement in APD is seen relatively quickly at $pp = 0.56$, soon after the unhedged model has reached its steady state. The improvement in ALO when using expansion hedges, for $pp = 0.5$, does not occur until after $7,500$ timesteps, well after the unhedged model has reached its steady state. However, the improvement in performance on ALO goes together with improved performance on APD which is attained at the same speed as the in the unhedged model.

If the speed of development of shared categories is not important, the two types of hedges would be useful in different types of situation, depending whether convergence or discriminatory power is more important. This might be dependent on, for example, the structure of the underlying environment. In the current simulation, objects are presented uniformly across the space. If objects were distributed non-uniformly, perhaps clumping in various regions of the

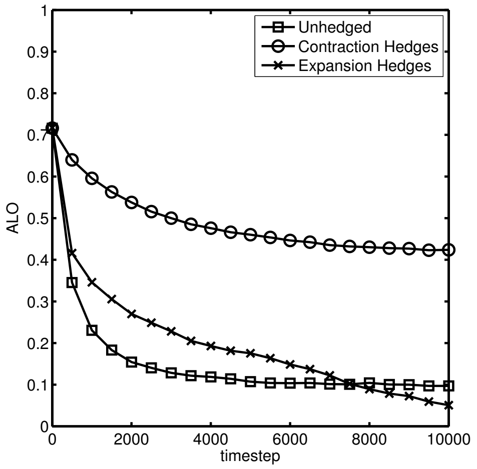

Figure 15. ALO vs time for models with no hedged assertions, contracted assertions and expanded assertions, for a prior probability of positive assertions $pp = 0.5$. Although the final value reached is lower when there is a high probability of using expansion hedges, the community of agents takes longer to reach that value, and may even reach a lower value still.

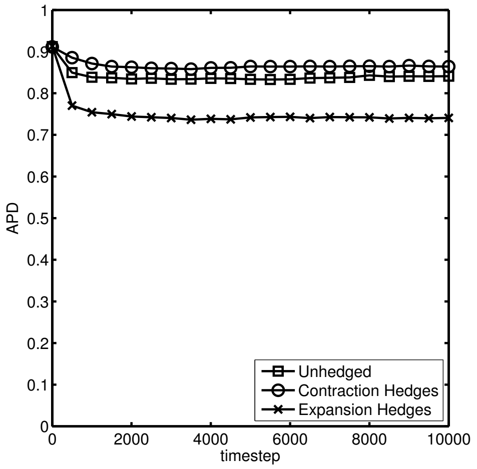

Figure 16. APD vs time for models with no hedged assertions, contracted assertions and expanded assertions, for a prior probability of positive assertions $pp = 0.5$. Lower APD is achieved with a high prior probability of asserting expansion hedges, in a similar timescale to the unhedged model

space, then perhaps the ability to discriminate precisely between different labels would be less important, since the environment provides that distinction naturally. Convergence to shared labels would then be more important.

If speed is important, using contraction hedges can still improve levels of convergence in a relatively short timeframe.

There are many parameters in the simulation that bear further investigation. The distribution of objects in the environment, as mentioned above, is likely to have an effect on performance again the two metrics. In the current simulations, hedge values of $v = 0.5$ and $h = 2$ are used. Increasing and decreasing these values could have an impact on performance, as would, perhaps, allowing agents to have difference values of v and h. The range of w allowed also affects performance. When $w = [0.01, 0.99]$, agents no longer achieve high levels of convergence at $pp > 0.55$ (results not shown). Other weight ranges may positively affect performance, however.

5 CONCLUSIONS

We have investigated the utility of hedged assertions in the development of a shared language, and shown that allowing agents to make hedged assertions improves the ability to develop common categories in two distinct ways. Firstly, using contraction hedges, i.e. words like ‘very’, allows improved levels of convergence to shared categories, whilst slightly improving the extent to which labels overlap. Secondly, using expansion hedges, or words like ‘quite’, enables the development of label sets that are more discriminatory of the environment and also have better levels of convergence. However, both these improvements come with a slower speed of development of shared labels. It may be possible to improve these speeds by tuning other parameters such as the age range of agents or the values of hedges used.

ACKNOWLEDGEMENTS

Martha Lewis gratefully acknowledges support from EPSRC Grant No. EP/E501214/1

REFERENCES

[1] P. Bosc, D. Dubois, A. HadjAli, O. Pivert, and H. Prade, ‘Adjusting the core and/or the support of a fuzzy set-a new approach to fuzzy modifiers’, in *Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International*, pp. 1–6. IEEE, (2007).
[2] H. Eyre and J. Lawry, ‘The evolution of vague categories within a conceptual space’. submitted.
[3] P. Gärdenfors, *Conceptual spaces: The geometry of thought*, The MIT Press, 2004.
[4] J. Hampton, ‘Conceptual combinations and fuzzy logic’, in *Concepts and Fuzzy Logic*, eds., R. Belohlavek and G. J. Klir, The MIT Press, (2011).
[5] J.A. Hampton, ‘Inheritance of attributes in natural concept conjunctions’, *Memory & Cognition*, **15**(1), 55–71, (1987).
[6] H. Kamp and B. Partee, ‘Prototype theory and compositionality’, *Cognition*, **57**(2), 129–191, (1995).
[7] J. Lawry, ‘A framework for linguistic modelling’, *Artificial Intelligence*, **155**(1-2), 1–39, (2004).
[8] J. Lawry and Y. Tang, ‘Uncertainty modelling for vague concepts: A prototype theory approach’, *Artificial Intelligence*, **173**(18), 1539–1558, (2009).
[9] D.N. Osherson and E.E. Smith, ‘On the adequacy of prototype theory as a theory of concepts’, *Cognition*, **9**(1), 35–58, (1981).
[10] E. Rosch, ‘Cognitive representations of semantic categories.’, *Journal of experimental psychology: General*, **104**(3), 192, (1975).
[11] E.E. Smith and D.N. Osherson, ‘Conceptual combination with prototype concepts’, *Cognitive Science*, **8**(4), 337–361, (1984).
[12] Luc Steels, ‘Why we need evolutionary semantics’, *KI 2011: Advances in Artificial Intelligence*, 14–25, (2011).
[13] Luc Steels, Tony Belpaeme, et al., ‘Coordinating perceptually grounded categories through language: A case study for colour’, *Behavioral and brain sciences*, **28**(4), 469–488, (2005).
[14] L.A. Zadeh, ‘Fuzzy sets*’, *Information and control*, **8**(3), 338–353, (1965).
[15] L.A. Zadeh, ‘A fuzzy-set-theoretic interpretation of linguistic hedges’, *Journal of Cybernetics*, (1972).

Balancing Parallel Two-Sided Assembly Lines with Ant Colony Optimisation Algorithm

Ibrahim Kucukkoc[a,b], **David Z. Zhang**[a], **Edward C. Keedwell**[a]

[a]College of Engineering, Mathematics and Physical Sciences, University of Exeter,
Harrison Building, EX4 4QF, Exeter, UK
[b]Department of Industrial Engineering, Balikesir University, Cagis Campus,
10145, Balikesir, Turkey
I.Kucukkoc@exeter.ac.uk, D.Z.Zhang@exeter.ac.uk, E.C.Keedwell@exeter.ac.uk

Abstract. Assembly lines are one of the most frequently used flow oriented production systems in industry. Although only a few researchers have studied them, two-sided assembly lines are usually utilised to produce high-volume large-sized products such as trucks and buses. In this study, more than one two-sided assembly line constructed in parallel are balanced simultaneously using a newly developed ant colony optimisation algorithm. To the best knowledge of the authors, the proposed method is the first attempt to solve the parallel two-sided assembly line balancing problem using an ant colony optimisation based algorithm. The proposed approach is also illustrated with examples from the literature to show the procedures of the algorithm.

Keywords: assembly line balancing; parallel two-sided assembly lines; ant colony optimisation; meta-heuristics; artificial intelligence.

1 INTRODUCTION

An assembly line is a sequence of workstations through which a set of tasks is processed. Assembly lines are used to assemble components into a final product and generally workstations are linked by a transportation system like a conveyor or moving belt [1].

Assembly line balancing (ALB) problem is one of the most common problems in industry and a classical Industrial Engineering problem. The main objective of balancing assembly lines is to increase the efficiency of the line by minimising required number of workstations (type I problem) or cycle time (type II problem) [2]. A task can be defined as the smallest work element which cannot be divided between two or more stations [3]. A set of tasks is performed at each workstation and each task has its own *processing time*. Due to technological and organisational conditions, *precedence constraints* must be satisfied in the assignment process [4, 5].

The sum of the completion times of tasks assigned to a work station is called as *workload* of this station. Usually, the cycle time is defined as a value which equals to the largest workload in an assembly line [6]. Hence, the production rate of the system is determined by cycle time [5, 7].

Assembly lines can be classified into two general groups: (i) one sided assembly lines, and (ii) two-sided assembly lines. While stations are utilised on only one side for one sided assembly lines, left and right sides are used to utilise stations for two-sided assembly lines. Two-sided assembly lines are chiefly used to produce large sized products like trucks and buses.

Although a large number of studies have been carried out in the literature on one sided assembly line balancing problem, the studies on two-sided assembly line balancing problem (TALBP) are very limited.

Two-sided assembly line balancing problem was defined by Bartholdi [8]. Bartholdi [8] discussed some theoretical properties of two-sided lines; and developed a first fit heuristic based computer program which embodies a balancing algorithm that emphasizes speed over accuracy for the interactive rapid refinement of solutions. Afterwards, meta-heuristics have been used to solve TALBP. Kim et al. [9, 10], Taha et al. [11], Purnomo et al. [12], and Rabbani et al. [13] developed different genetic algorithms while Baykasoglu and Dereli [14], and Simaria and Vilarinho [15] developed ant colony optimisation (ACO) based algorithms. The study belongs to Baykasoglu and Dereli [14] is one of the first attempts to solve TALBPs using ant colony based heuristic. As different from some other studies, Simaria and Vilarinho [15] employed two ants that work concurrently - one at each side of the line - to build a balancing solution. Ozcan and Toklu [16], and Ozcan [17] implemented simulated annealing algorithms; Ozcan and Toklu [18] proposed tabu search algorithm; Chutima and Chimklai [19] proposed particle swarm optimisation algorithm while Ozbakir and Tapkan [20, 21] developed bees algorithms to solve TALBP. Some exact solution approaches have also been applied to TALBP by Hu et al. [22, 23], and Wu et al. [24]. However, none of these studies considered more than one line. Furthermore, this is the unique study in the literature which applies ACO on any kind of parallel two-sided assembly line balancing problem.

In this study, more than one two-sided assembly line balancing problem is balanced simultaneously. This problem is known as the *parallel two-sided assembly line balancing problem (PTALBP)* and first (and only) studied by Ozcan et al. [25]. They described the problem and proposed a tabu search algorithm to solve PTALBP.

Simple assembly line balancing problem (SALBP), which is the simplest version of assembly line balancing problems, is an NP-hard class of combinatorial problem [26]. Since PTALBP is a much more complex version of SALBP, it is also NP-Hard, which means that it is difficult to obtain an optimal solution when the problem size increases. Because, the solution space

---

[b] ikucukkoc@balikesir.edu.tr

will grow exponentially as the number of tasks increase [24]. It is the major reason why a considerable amount of researches in the literature strives to develop heuristics and meta-heuristics instead of exact algorithms to solve ALB problems.

Therefore, a new ant colony optimisation algorithm has been developed to solve PTALBP based on this motivation. To the best knowledge of the authors, the proposed algorithm is the first attempt to solve PTALBP with an ACO based approach in the literature. The main contribution of this research is proposed new and first ACO algorithm with different pheromone release strategy to solve PTALBP.

2 PROBLEM STATEMENT

The main objective of assembly line balancing is to minimise the sum of the differences between the cycle time and the individual workloads, which means minimising the total idle time of the line by minimising the number of required stations or the cycle time.

A parallel, two-sided assembly line balancing problem is one in which more than one two-sided assembly lines are constructed in parallel. The main aim is assigning tasks to the workstations by considering some specific constraints like technological priorities, capacity constraints, zoning constraints, and positional constraints [25].

Different product models are produced on different two-sided assembly lines $(h = 1, \dots, H)$ and each product model $(m = 1, \dots, M)$ has its own set of tasks $(i = 1, \dots, N_h)$. Each task requires a certain amount of time, which is called task time (t_{ih}), to be processed. Predefined precedence relationships are considered to perform these tasks. P_{ih} represents the set of predecessors of task i in line h. Tasks are performed in a series of workstations $(k = 1, \dots, S)$ utilised on parallel lines, considering cycle time (C_h) which is predefined by line manager [25]. q_k symbolises the queue number in which station k is utilised.

The cycle time of each line may be different from each other. In that case, a common time should be used while assigning tasks to the stations, in each cycle. Gokcen et al. [1] used least common multiple (LCM) based approach for this issue. Please refer to the studies of Gokcen et al. [1], and Ozcan et al. [25] for further information on LCM approach.

Interference is an important issue that must be taken into account while balancing two-sided assembly lines. To avoid interference, tasks which have precedence relationship with each other must be assigned by considering completion time of previously assigned task. To give an example about that, let us consider P_{f2} as set of predecessors of task f (in Figure 1). Task f can only be initialised after completion of task e, which is assigned on the other side of the line (to its mated workstation). The shaded rectangles in the figure represent idle times of the line.

The merging of stations is one of the useful advantages of parallel two-sided assembly lines. As can be seen from Figure1, operator 2 performs tasks on both lines. First, Operator 2 completes task e on the left side of the line II and then completes tasks c, and d respectively on the right side of the line I. Thus, operators which are assigned to interval of two lines can be used effectively to increase the efficiency of the lines.

In this study, we consider two parallel two-sided assembly lines $(H = 2)$, so the number of different product models equals to two $(M = 2)$.

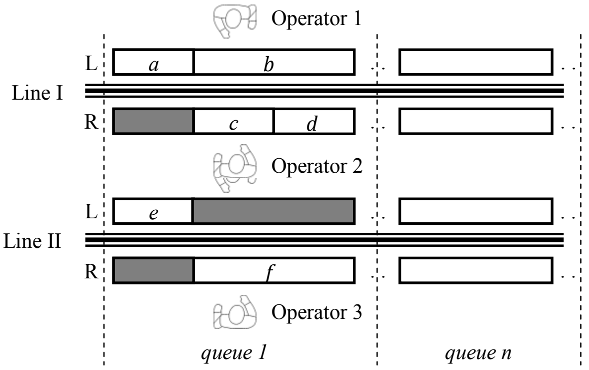

Figure 1 Representation of parallel two-sided assembly lines (adapted from [25]).

The assumptions/constraints considered in the study are as follows:

- Only one product model is assembled on each line $(H = M)$,
- Task times are known and deterministic,
- Each task must be assigned to exactly one workstation,
- Cycle time is larger than total workload of any workstation,
- Each product model has its own set of precedence relationships,
- Tasks can be assigned only a predetermined side of the line (left - L, right - R, or either - E),
- A task can only be started if all of its predecessor tasks have been assigned and completed,
- Interval stations are merged if the efficiency of an interval station is less than 75%.

The objective function of the proposed method is given in Equation 1.

$$\min\left(\sum_{k=1}^{S} z_k + \varepsilon \max_{k=1,\dots,S}\{q_k\}\right) \tag{1}$$

where z_k is a binary variable and ε is a user defined weighting factor;

$$z_k = \begin{cases} 1 & if\ station\ k\ is\ utilised \\ 0 & otherwise \end{cases} \tag{2}$$

The first term of the objective function corresponds to the total number of opened workstations while the second term represents the total length of the line. The weighting factor (ε) allows the user to decide on the significance of these two major aims.

If ε is selected larger than 1, it means that total length of the line is more important than the utilised number of stations to perform assigned tasks.

As can be observed from the Figures 1 and 2, the lines have been divided into different *queue*s and *qzone*s. qzone value helps the algorithm in holding the information for any specific task in which section of the line is the task assigned. So, this value will be used for pheromone release strategy, as a new feature of the algorithm developed in this research.

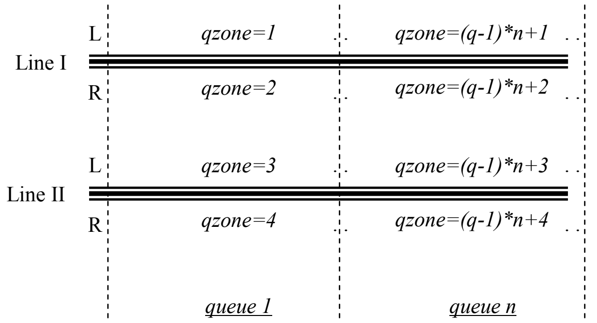

Figure 2 Dividing parallel two-sided assembly lines into different qzones.

3 PROPOSED METHOD

Meta-heuristic algorithms are widely used to solve engineering design problems; especially to deal with NP-Hard combinatorial ones.

The ant colony optimisation (ACO) proposed by Dorigo et al. [27], is one of number of nature inspired algorithms. They developed an ant system (AS) meta-heuristic (initial form of ACO) to solve small-sized travelling salesman problem (TSP). Since then, several researchers carried out a substantial amount of research in ACO algorithm, which demonstrates a better performance than AS. ACO algorithm has some mechanisms, which mimics the behaviour of the real ant colonies.

One of the main application areas of ACO algorithm is assembly line balancing problem. Some researchers have found that ant techniques are able to solve various types of ALB problems, but not PTALBP.

To our knowledge, the first technique that uses concepts derived from ant colony optimisation in this domain was applied by McMullen and Tarasewich [28] to mixed-model assembly line balancing problem with parallel workstations. Then, Vilarinho and Simaria [29], and Yagmahan [30] proposed different ACO algorithms for different type of mixed-model assembly line balancing problems., Zhang et al. [31], Fattahi et al. [32], and Sulaiman et al. [33] applied ant colony optimisation algorithm for straight assembly lines with various objective functions. Among them, Bautista and Pereira [34], and Chica et al. [35, 36] considered also space constraints.

Baykasoglu and Dereli [37], Khaw and Ponnambalam [38], and Sabuncuoglu et al. [39] addressed U-shaped assembly line balancing problem via ant colony optimisation algorithm. Moreover, Baykasoglu and Dereli [14], and Simaria and Vilarinho [15] balanced two-sided assembly lines while Baykasoglu et al. [40], and Ozbakir et al. [41] optimised balanced parallel assembly lines using ACO based approaches.

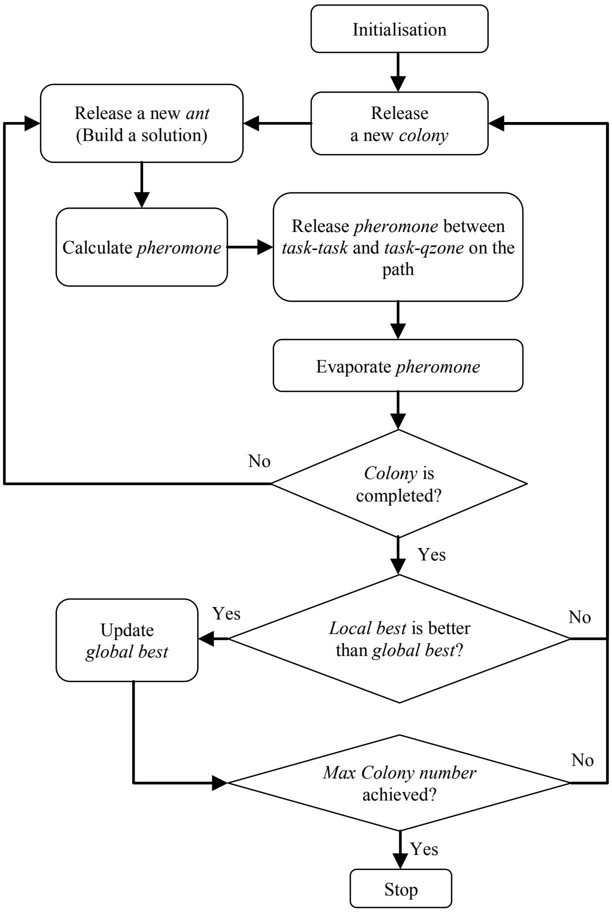
Initialisation
Release a new *ant* (Build a solution)
Release a new *colony*
Calculate *pheromone*
Release *pheromone* between *task-task* and *task-qzone* on the path
Evaporate *pheromone*
No
Colony is completed?
Yes
Update *global best*
Yes
Local best is better than *global best*?
No
Max Colony number achieved?
No
Yes
Stop

Figure 3 Flowchart of proposed ACO.

However, as mentioned before, none of these researchers addressed PTALBP problem to solve via an ACO based approach.

Figure 3 gives a flowchart of proposed ant colony optimisation approach in this study. The proposed algorithm starts with initialisation of pheromones. A new colony is released and different solutions (paths) are obtained by each ant in the colony. The basic idea is selection of tasks to be added to the current workstation by artificial ants. Pheromone level determines the probability of a task being selected by an ant. Pheromones, a measure of each path's relative desirability, are calculated according to the quality of the drawn path by each ant.

In the algorithm a new pheromone releasing strategy has been used instead of a heuristic search. So, two types of pheromone have been released by each ant according to the quality of the drawn path: (i) between *task* and *last assigned task*, and (ii) between *task* and *qzone* number (see Figure 2 for the explanation of qzone). A constant value of pheromone is evaporated after each tour. When a colony is completed their tour, global best solution is updated if a better solution is found, and double pheromone is laid to the edges of global best solution. The algorithm continues until all colonies complete their tour and stops when a predetermined maximum colony (*Max Colony*) number has been exceeded.

Transition rule and pheromone evaporation rule are given in Equations 3 and 4, respectively.

So, the probability for ant k to select task i after task j in qzone z while in its t^{th} tour is:

$$p_{ijz}^{k}(t)=\frac{[\tau_{ij}(t)]^{\alpha}[\eta_{iz}(t)]^{\beta}}{\sum_{h\epsilon J_i^k, y\epsilon Z_i^k}[\tau_{ih}(t)]^{\alpha}[\eta_{iy}(t)]^{\beta}} \tag{3}$$

Each ant k maintains a tabu list in memory that defines the sets:

J_i^k: tasks still to be assigned when at task i,

Z_i^k: qzones still to be utilised when at task i.

The amount of virtual pheromone between *task - last assigned task* is represented with $\tau_{ij}(t)$. As a new feature of the algorithm, the pheromone amount between *task - qzone* is represented with $\eta_{iz}(t)$.

After the completion of a tour, each ant k lays a quantity of pheromone $(100/Q)$ on both *task-task*, and *task-qzone* matrixes. Q is the value which represents the quality of the acquired solution, and obtained by objective function, which is given in Equation 1. Evaporated pheromone level is calculated via Equation (4).

$$\tau=(1-\rho)\tau \tag{4}$$

Pheromone update rules for *task-task* matrix, and *task-qzone* matrix are given in Equations 5 and 6, respectively.

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\Delta\tau_{ij}(t) \tag{5}$$

$$\tau_{iz}(t+1)=(1-\rho)\tau_{iz}(t)+\Delta\tau_{iz}(t) \tag{6}$$

$$\Delta\tau_{ij}(t)=\Delta\tau_{iz}(t)=\frac{100}{Objective\ Function\ Value} \tag{7}$$

Pseudo code of building a balancing solution procedure is given below (see Figure 4). Each ant draws a path using this code to build a balancing solution.

```
Select first line
Start from left side
While (there is (are) unassigned task(s))
    Determine available tasks (in terms of precedence,
    capacity, and other constraints)
    Select an available task using pheromone information
    Assign selected task to current workstation
    Increase station workload as task time
        If ($st(k) > st(\underline{k})$) then
            Change side (left or right)
        End if
        If (both sides do not have enough capacity to
        assign available tasks) then
            Select other line
            Start from left side
        End if
        Update unassigned tasks list
End while
```

Figure 4 Pseudo code of balancing a solution by each ant.

In the code, $st(k)$ means workload of current workstation while $st(\underline{k})$ represents workload of its mated workstation [15].

While allocating tasks to line I, if both sides do not have enough capacity to assign available tasks from line I (product model 1), efficiency of right side workstation is checked whether more tasks can be assigned from line II (product model 2). If yes, tasks are assigned from line II until right side workstation gets full, to decrease idle times.

4 EXAMPLES

Two examples are given to enable understanding of the proposed method and its main components. For this purpose, well-known problems from the literature are used to obtain different product models and solved using the proposed algorithm.

The algorithm is coded in Java SE 7u4 and executed on a 3.1 GhZ Intel Core i5-2400 CPU computer. After max number of colonies is achieved, algorithm is terminated and best solution is taken. Additionally, best solutions are recorded after completion of each ant's tour in order to analyse the efficiency of the algorithm.

The parameters used in the study are determined through a set of preliminary experimentations by authors and given in Table 1. These experiments are performed to improve the quality of obtained solutions.

Table 1 Parameters used in the study for illustrated examples.

Parameter	Example 1	Example 2
α	0.1	0.1
β	0.3	0.3
ρ	0.1	0.1
ε	0.5	0.5
Initial pheromone level	8	20
Number of ants in each colony	10	10
Total number of colonies	10	20

4.1. Example 1

In the first example, two product models which have same precedence relationships and task times - they also can be thought of as the same product - produced on two parallel lines, one on each assembly line. Data of precedence relationships, task times, and preferred operation directions (side) for this problem is collected from the study of Kim et al. [9] and represented in Figure 5 (problem P9). Cycle time is accepted as 6 minutes for both lines.

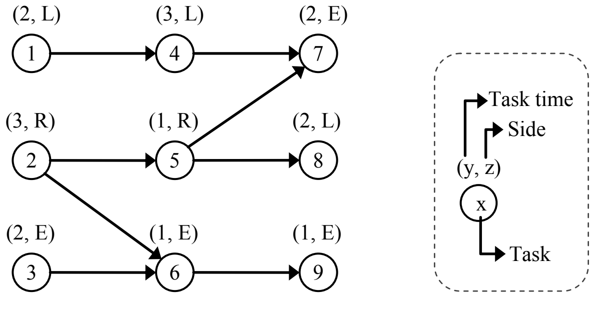

Figure 5 Precedence relationship diagram for illustrated example 1 (adapted from [9]).

Table 2 shows obtained best solution for given example. As can be seen from the table, 6 workstations are utilised to perform 18 different tasks that belong to two product models. Total length of the line is 2 units. Four workstations are charged full capacity while two workstations in queue 2 have 1 minute, and 1 minute idle times, individually.

Table 2 Obtained best solution for illustrated example 1.

Line	Side	Queue 1		Queue 2	
		Tasks	Workload	Tasks	Workload
1	Left	1,4,6	6	7,8,9	5
	Right	2,3,5	6	-	-
2	Left	3,1,8	6	4,7	5
	Right	2,5,6,9	6	-	-

Figure 6 illustrates the convergence of the algorithm after releasing each ant. Best solutions are given in the figure from ant one to ant 100. Variation of station number and line length values can also be investigated from the figure. As can be observed from the figure, the behaviour of the algorithm is quite promising.

The proposed ACO finds the optimal number of stations within 2 seconds and gets the best solution 6 at earlier colonies.

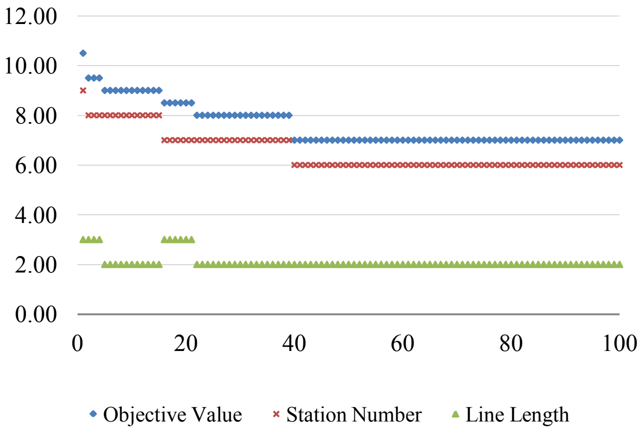

Figure 6 Convergence of objective function value for illustrated example 1.

4.2. Example 2

For second example, two different product models that have same precedence relationships but different task times are assembled on two parallel lines (one on each line). Precedence relationship diagram for two product models is taken from a well-known test problem (P12) studied by Kim et al. [9], and given in Figure 7.

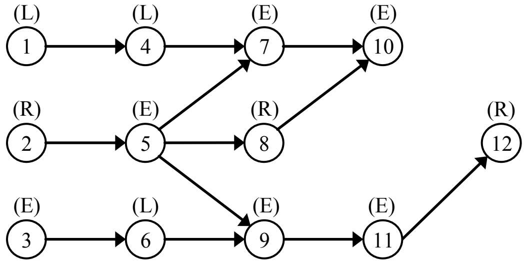

Figure 7 Common precedence relationship diagram for illustrated example 2 (adapted from [9]).

In the precedence relationship diagram (given above), preferred operation directions (sides) are also shown over nodes. Task times are not given on the graph since different task times are executed randomly for two different product models. Table 3 exhibits randomly generated task times for two different product models (1, and 2). Cycle time is considered as 9 minutes for both lines.

Table 3 Task times for illustrated example 2.

Product Model 1		Product Model 2	
Task No	Time (min)	Task No	Time (min)
1	1	1	2
2	3	2	1
3	2	3	4
4	4	4	3
5	4	5	7
6	3	6	6
7	2	7	2
8	7	8	5
9	3	9	3
10	5	10	2
11	4	11	2
12	3	12	7

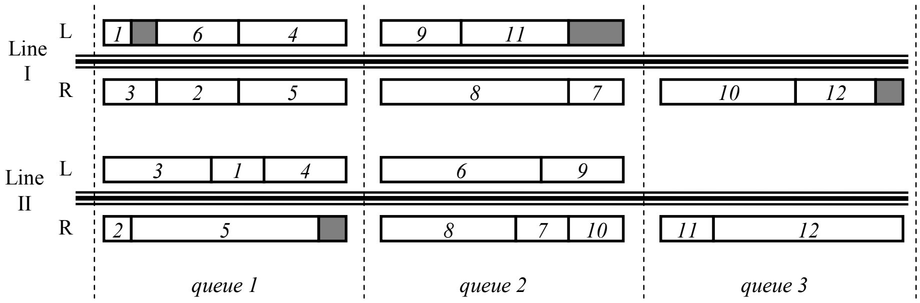

Figure 8 Obtained best solution for illustrated example 2.

When the program runs, it finds the solution represented in Figure 8 in few seconds and at earlier stages of the program. As can be investigated from Figure 8, 10 workstations are utilised to perform 18 different tasks that belong to two different product models. Total required line length is 3 units. Six workstations are loaded full capacity while 3 workstations have 1 minute idle time and one workstation has 2 minutes idle time.

To investigate the convergence of the algorithm, change in required total number of workstations, total line length, and objective function value by iterations are given in Figure 9.

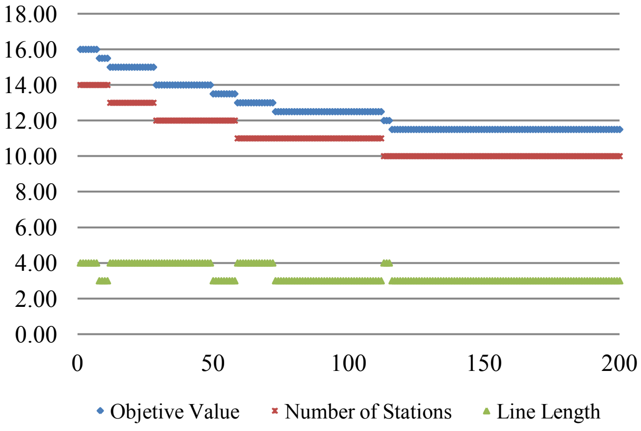

Figure 9 Convergence of objective function value for illustrated example 2.

5 DISCUSSIONS

The proposed method is illustrated with two different small-sized examples, which have 18 and 24 tasks, respectively. However, since PTALBP is a much more complex version of SALBP, it is also NP-Hard, which means that it is difficult to obtain an optimal solution when the problem size increases. As the nature of NP-Hard problems, solution space will grow exponentially as the number of tasks increases [24]. So, solving this type of problems in a reasonable time using an exact solution method is quite hard.

The proposed algorithm has the advantage of different pheromone releasing strategy between *task – qzone*. Using this sort of trail helps ants to find their way more efficiently in such complex problems. Because, PTALBP is much more complex form of SALBP and operation sides that tasks are assigned in to are also important as well as the last assigned task. However, a heuristic algorithm may be required to be integrated with developed ant colony algorithm to solve medium and large sized parallel two-sized assembly line balancing problems. Furthermore, more than one heuristic search algorithms can be integrated with proposed ACO algorithm. Thus, each ant may have different behaviours to select next available task.

The illustrated examples use same cycle times for parallel lines. If we take into account different cycle time situations for parallel lines, task times should be modified and a common cycle time should be accepted as common cycle time to adapt LCM approach proposed by Gokcen et al. [1].

Some parameters vary between example 1 and example 2 in order to search solution space more efficiently. For example, initial pheromone level is selected 8 in example 1, while 20 in example 2. Similarly, 10 colonies are employed in example one whilst 20 colonies are charged in example 2. The reason is that, total number of tasks for example 2 is larger than example 1, and solution space grows exponentially as the problem size increases.

None of the workstations are merged in the illustrated examples however it is most likely to be occurred in medium and large sized problems.

6 CONCLUSION AND FUTURE RESEARCH

In this paper, an ant colony optimisation algorithm has been proposed for solving type-I parallel two-sided assembly line balancing problem.

The main purpose of this research is to explore the potential of ACO approach and show how more than one two-sided assembly line, which is constructed in parallel, is balanced together using an ant colony optimisation based approach. To the best knowledge of the authors, the proposed algorithm is the first attempt to solve PTALBP with an ACO based approach.

The proposed ACO algorithm makes use of the trail information which is deposited between the *task* and *last assigned task*, and the *task* and *qzone*, which is the position in which the task is allocated. Objective function of the proposed method aims to minimise total number of workstations while considering total length of the line. A weighting factor, which determines the significance of line length, is also integrated to the objective function.

Two examples are given from the literature and solved using proposed ACO algorithm. Results of the examples show that proposed method performs remarkably well. However, to assess the performance and efficiency of the algorithm, a set of benchmark problems can be solved with the proposed approach for future research directions. Obtained results can also be compared with existing tabu search algorithm in the literature.

Some additional constraints such as positive and negative zoning constraints, synchronous tasks constraints, and positional constraints can also be considered for future researches.

7 ACKNOWLEDGEMENTS

First author wishes to thank Balikesir University and all authors wish to thank University of Exeter for their financial supports which helps to perform this research. The authors also would gratefully like to thank the anonymous referees whose valuable suggestions lead to an improved organisation of this paper.

8 REFERENCES

[1] Gokcen, H., Agpak, K., and Benzer, R., (2006) "Balancing of parallel assembly lines", *International Journal of Production Economics*, **103**, No 2, pp600-609.

[2] Kucukkoc, I. and Yaman, R., (2013) "A new hybrid genetic algorithm to solve more realistic mixed-model assembly line balancing problem", *International Journal of Logistics Systems and Management*, **14**, No 4, pp (in press).

[3] Baybars, I., (1986) "An Efficient Heuristic Method for the Simple Assembly Line Balancing Problem", *International Journal of Production Research*, **24**, No 1, pp149-166.

[4] Sarker, B.R. and Shanthikumar, J.G., (1983) "A Generalized-Approach for Serial or Parallel Line Balancing", *International Journal of Production Research*, **21**, No 1, pp109-133.

[5] Simaria, A.S., (2006) *Assembly Line Balancing - New Perspectives and Procedures*, in *Departamento de Economia, Gestao e Engenharia Industrial*, PhD Thesis, Universidade de Aveiro.

[6] Kara, Y., Ozguven, C., Yalcin, N., and Atasagun, Y., (2011) "Balancing straight and U-shaped assembly lines with resource dependent task times", *International Journal of Production Research*, **49**, No 21, pp6387-6405.

[7] Darel, E.M. and Cother, R.F., (1975) "Assembly line sequencing for model mix", *International Journal of Production Research*, **13**, No 5, pp463-477.

[8] Bartholdi, J.J., (1993) "Balancing 2-Sided Assembly Lines - a Case-Study", *International Journal of Production Research*, **31**, No 10, pp2447-2461.

[9] Kim, Y.K., Kim, Y.H., and Kim, Y.J., (2000) "Two-sided assembly line balancing: a genetic algorithm approach", *Production Planning & Control*, **11**, No 1, pp44-53.

[10] Kim, Y.K., Song, W.S., and Kim, J.H., (2009) "A mathematical model and a genetic algorithm for two-sided assembly line balancing", *Computers & Operations Research*, **36**, No 3, pp853-865.

[11] Taha, R.B., El-Kharbotly, A.K., Sadek, Y.M., and Afia, N.H., (2011) "A Genetic Algorithm for solving two-sided assembly line balancing problems", *Ain Shams Engineering Journal*, **2**, No 3-4, pp227-240.

[12] Purnomo, H.D., Wee, H.M., and Rau, H., (2013) "Two-sided assembly lines balancing with assignment restrictions", *Mathematical and Computer Modelling*, **57**, No 1-2, pp189-199.

[13] Rabbani, M., Moghaddam, M., and Manavizadeh, N., (2012) "Balancing of mixed-model two-sided assembly lines with multiple U-shaped layout", *International Journal of Advanced Manufacturing Technology*, **59**, No 9-12, pp1191-1210.

[14] Baykasoglu, A. and Dereli, T., (2008) "Two-sided assembly line balancing using an ant-colony-based heuristic", *International Journal of Advanced Manufacturing Technology*, **36**, No 5-6, pp582-588.

[15] Simaria, A.S. and Vilarinho, P.M., (2009) "2-ANTBAL: An ant colony optimisation algorithm for balancing two-sided assembly lines", *Computers & Industrial Engineering*, **56**, No 2, pp489-506.

[16] Ozcan, U. and Toklu, B., (2009) "Balancing of mixed-model two-sided assembly lines", *Computers & Industrial Engineering*, **57**, No 1, pp217-227.

[17] Ozcan, U., (2010) "Balancing stochastic two-sided assembly lines: A chance-constrained, piecewise-linear, mixed integer program and a simulated annealing algorithm", *European Journal of Operational Research*, **205**, No 1, pp81-97.

[18] Ozcan, U. and Toklu, B., (2009) "A tabu search algorithm for two-sided assembly line balancing", *International Journal of Advanced Manufacturing Technology*, **43**, No 7-8, pp822-829.

[19] Chutima, P. and Chimklai, P., (2012) "Multi-objective two-sided mixed-model assembly line balancing using particle swarm optimisation with negative knowledge", *Computers & Industrial Engineering*, **62**, No 1, pp39-55.

[20] Ozbakir, L. and Tapkan, P., (2010) "Balancing fuzzy multi-objective two-sided assembly lines via Bees Algorithm", *Journal of Intelligent & Fuzzy Systems*, **21**, No 5, pp317-329.

[21] Ozbakir, L. and Tapkan, P., (2011) "Bee colony intelligence in zone constrained two-sided assembly line balancing problem", *Expert Systems with Applications*, **38**, No 9, pp11947-11957.

[22] Hu, X.F., Wu, E.F., and Jin, Y., (2008) "A station-oriented enumerative algorithm for two-sided assembly line balancing", *European Journal of Operational Research*, **186**, No 1, pp435-440.

[23] Hu, X.F., Wu, E.F., Bao, J.S., and Jin, Y., (2010) "A branch-and-bound algorithm to minimize the line length of a two-sided assembly line", *European Journal of Operational Research*, **206**, No 3, pp703-707.

[24] Wu, E.F., Jin, Y., Bao, J.S., and Hu, X.F., (2008) "A branch-and-bound algorithm for two-sided assembly line balancing", *International Journal of Advanced Manufacturing Technology*, **39**, No 9-10, pp1009-1015.

[25] Ozcan, U., Gokcen, H., and Toklu, B., (2010) "Balancing parallel two-sided assembly lines", *International Journal of Production Research*, **48**, No 16, pp4767-4784.

[26] Wee, T.S. and Magazine, M.J., (1982) "Assembly line balancing as generalized bin packing", *Operations Research Letters*, **1**, No 2, pp56-58.

[27] Dorigo, M., Maniezzo, V., and Colorni, A., (1996) "Ant system: Optimization by a colony of cooperating agents", *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, **26**, No 1, pp29-41.

[28] McMullen, P.R. and Tarasewich, P., (2003) "Using ant techniques to solve the assembly line balancing problem", *Iie Transactions*, **35**, No 7, pp605-617.

[29] Vilarinho, P.M. and Simaria, A.S., (2006) "ANTBAL: an ant colony optimization algorithm for balancing mixed-model assembly lines with parallel workstations", *International Journal of Production Research*, **44**, No 2, pp291-303.

[30] Yagmahan, B., (2011) "Mixed-model assembly line balancing using a multi-objective ant colony optimization approach", *Expert Systems with Applications*, **38**, No 10, pp12453-12461.

[31] Zhang, Z.Q., Cheng, W.M., Tang, L.S., and Zhong, B., (2007) "Ant algorithm with summation rules for assembly line balancing problem", *Proceedings of 14th International Conference on Management Science & Engineering*, **Vols 1-3**, pp369-374.

[32] Fattahi, P., Roshani, A., and Roshani, A., (2011) "A mathematical model and ant colony algorithm for multi-manned assembly line balancing problem", *International Journal of Advanced Manufacturing Technology*, **53**, No 1-4, pp363-378.

[33] Sulaiman, M.N.I., Choo, Y.H., and Chong, K.E., (2011) "Ant Colony Optimization with Look Forward Ant in Solving Assembly Line Balancing Problem", *3rd Conference on Data Mining and Optimization (DMO)*, pp115-121.

[34] Bautista, J. and Pereira, J., (2007) "Ant algorithms for a time and space constrained assembly line balancing problem", *European Journal of Operational Research*, **177**, No 3, pp2016-2032.

[35] Chica, M., Cordon, O., Damas, S., and Bautista, J., (2010) "Multiobjective constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search", *Information Sciences*, **180**, No 18, pp3465-3487.

[36] Chica, M., Cordon, O., Damas, S., and Bautista, J., (2011) "Including different kinds of preferences in a multi-objective ant algorithm for time and space assembly line balancing on different Nissan scenarios", *Expert Systems with Applications*, **38**, No 1, pp709-720.

[37] Baykasoglu, A. and Dereli, T., (2009) "Simple and U-Type Assembly Line Balancing by Using an Ant Colony Based Algorithm", *Mathematical & Computational Applications*, **14**, No 1, pp1-12.

[38] Khaw, C.L.E. and Ponnambalam, S.G., (2009) "Multi-Rule Multi-Objective Ant Colony Optimization for Straight and U-Type Assembly Line Balancing Problem", *2009 IEEE International Conference on Automation Science and Engineering*, pp177-182.

[39] Sabuncuoglu, I., Erel, E., and Alp, A., (2009) "Ant colony optimization for the single model U-type assembly line balancing problem", *International Journal of Production Economics*, **120**, No 2, pp287-300.

[40] Baykasoglu, A., Ozbakir, L., Gorkemli, L., and Gorkemli, B., (2009) "Balancing Parallel Assembly Lines via Ant Colony Optimization", *CIE: 2009 International Conference on Computers and Industrial Engineering, Vols 1-3*, pp506-511.

[41] Ozbakir, L., Baykasoglu, A., Gorkemli, B., and Gorkemli, L., (2011) "Multiple-colony ant algorithm for parallel assembly line balancing problem", *Applied Soft Computing*, **11**, No 3, pp3186-3198.

Ecological Optimization using BBO Metaheuristic

Daoudi Mourad

Computer Science Department, LSI Laboratory, USTHB
BP 32 16111 El Alia, Bab-Ezzouar,
Algiers, Algeria
m-daoudi@usthb.dz

Abstract. The ecological conservation problem for preserving species and their habitats was formulated as an optimization problem of maximal covering species problem in order to find the maximal number of species while limiting the number of selected parcels to P. It is a combinatorial optimization problem NP-Hard, and thus intractable with classical methods when data is very large. Metaheuristics offer an alternative to solve this type of problem. We applied a recently-developed Biogeography Based Optimzation metaheuristic. Tests were performed, and comparisons are made with another method using Harmony Searh metaheuristic.

Keywords - Ecology, Biogeography Based Optimization, Maximal Covering Species Problem, Optimization, Harmony Search.

1 INTRODUCTION

The conservation of biological diversity has become a major challenge for different reasons like environmental (stability of ecosystems, geochemical cycles) and socio-economic reasons (food, agriculture, fisheries, industrial application) and also for evident ethical reasons of life intrinsic value. Location choice is always an important concern in establishing biological reserves.

A large amount of literature addresses the reserve site selection problem. Methods for systematically selecting sites (land units) for a nature reserve were first devised more than 20 years ago, beginning with the pioneering work of Kirkpatrick [1, 2]. These methods sought to identify reserve systems in which biodiversity, measured quantitatively, would be represented at desired levels.

The models termed Species Set Covering Problem (SSCP) and Maximal Covering Species Problem (MCSP) are counterparts of location models [3] respectively developed in 1971 and 1974 and termed Location Set Covering Problem [4] and Maximal Covering Location Problem [5]. They aim at delineating nature reserves for protecting either all species or as many species as possible. The motivation of these models is to conserve biological diversity by setting land aside for the creation or enlargement of nature reserves that protect the key habitats and species.

The SSCP model was first developed by Possingham et al. (1993) [6]. It is to choose the least number of land sites (parcels) in such a way that each species is protected, i.e. represented in at least one parcel. The MCSP model, in contrast, aims at maximizing the number of species for a given number of land sites (parcels) to be selected or for a given number of parcels that can be selected, or a given amount of budget.

The mathematical formulation of the SSCP model is as follows:

$$Min \sum_{j\epsilon J} Xj \qquad (1)$$

$$s.t \ \sum_{j\epsilon Mi} Xj \ \geq 1, \quad all \ \ i\epsilon I \qquad (2)$$

where j and J are the index and set of land parcels, respectively; i and I are the index and set of species, respectively; M_i is the set of parcels j that include species i ; and X_j is a binary variable for parcel selection (it has 1 if parcel j is selected, and has 0 otherwise). The objective (1) minimizes the number of sites selected, and the constraint set (2) requires the selection of at least one site containing each species.

Rather than minimizing the necessary land parcels to protect all the endangered species, the MCSP model, first formulated in 1996 by Church et al. [7] and Camm et al. [8], focuses on maximizing the protected number of species for a given number of land sites selected. It is to find the maximal number of species while limiting the number of selected parcels to P.

The mathematical formulation of the MCSP model is as follows:

$$\text{Max} \sum_{i\epsilon I} Yi \qquad (3)$$

$$\text{s.t.} \sum_{j\epsilon Mi} Xj \ \geq \ Yi \ , \quad \text{all i}\epsilon\text{I} \qquad (4)$$

$$\sum_{j\epsilon J} Xj = P \qquad (5)$$

where Y_i is a binary variable for species covering (it has 1 if species Y_i is covered, and has 0 otherwise). The objective (3) maximizes the number of species represented, whereas constraint set (4) enforces the condition that a species is represented only if a site containing that species is selected, and constraint (5) limits the number of sites that may be selected.

The MSCP model was especially tackled by various algorithms [9-11]. It is a combinatorial optimization problem NP-Hard [12] and thus intractable with classical methods when data is very large. Metaheuristics offer an alternative to solve this type of problem. In our approach to tackle the MSCP problem, we use a recently-developed Biogeography Based Optimization algorithm (BBO) [13]. Comparisons of our results with those of another meta-heuristic algorithm, Harmony Search (HS) [14] are made.

The rest of the paper is as follows: BBO algorithm is outlined in section 2, with our contribution to solve MCSP problem. Experimental results are reported in Section 3. Comparisons with HS are also made. Section 4 is conclusion.

2 SOLVING MCSP

2.1. BBO overview

Biogeography studies the geographical distribution of biological organisms. It is due to the work of Alfred Wallace [15] and Charles Darwin [16] in the 19th century. This work had a descriptive and historical aspect. In 1960, Robert MacArthur and Edward Wilson have developed a mathematical model for the biogeography [17]. They were interested in the distribution of habitat species. A habitat represents any living space isolated from other spaces. The mathematical model that was developed, describes how some species migrate from one habitat to another. Habitats that are favorable to the residence of biological species are called high habitat suitability index (HSI). The parameters influencing the HSI may be rainfall, crop diversity, diversity of terrain ... etc. The variables describing the habitability are called suitability index variables (SIV).

Habitats are characterized by the following:

- A habitat with a high HSI tends to have a high number of species, while those with a low HSI tends to have fewer species.
- The habitats with high HSI are characterized by a high rate of emigration and low rate of immigration because they are saturated with species.
- The habitats with low HSI have a high immigration rate and a low emigration rate.

Immigration should result in the modification of the HSI of the habitat. If the HSI habitat remains too long without improving, the species that live there tend to disappear.

Dan Simon introduced in 2008 a metaheuristic based on biogeography [13]. It uses the following analogies:

- A solution is analogous to a habitat
- The quality of the solution (fitness) is analogous to the HSI.
- The variables defining the solution are analogous to SIVs.
- A good solution is analogous to a habitat with a high HSI, and thus with a high number of species, a high rate of emigration and a low immigration rate.
- A bad solution is analogous to a habitat with a low HSI, a low number of species, a low emigration rate and a high immigration rate.
- A good solution tends to share characteristics with a bad solution to improve it (migration of SIVs). This is analogous to the migration of species between habitats. Sharing characteristics does not involve change in the characteristics of good solutions, because migration deals only with a sample of species, so that it does not affect the habitat.
- The bad solutions accept the characteristics of good solutions in order to improve their quality. This is analogous to the bad habitat that accepts immigration of species from other habitats.

Improving the population is the way to solve problems in heuristic algorithms. The method to generate the next generation in BBO is by immigrating solution features to other habitats, and receiving solution features by emigration from other habitats.

As described in the migration algorithm given below, suppose we have a population of candidate solutions to a problem, represented by vectors (Habitats Hi, i=1…n). Each element of the vector is considered as an SIV value. In the migration process, the characteristics of good solutions replace the worst ones when using the immigration rate λ and the emigration rate μ, according to a probability of change P_{mod}.

In the migration process, when a solution is selected to be changed, we use the rate of immigration λ to decide whether an SIV will be changed. In this case, we use the emigration rate μ to decide which good solution in the habitat will migrate its SIV.

In the mutation process, mutation is performed for the whole or part of the population in a manner similar to the mutation in genetic algorithms (GAs). Changes in a habitat may arise. These modifications will change the HSI of the habitat. This is modeled in BBO when muting with a certain probability. Mutated habitats are those having a low HSI. This mutation introduces diversity and encourages poor habitats to improve.

In evolutionary strategies, global recombination is used to create new solutions, while BBO migration is used to change existing solutions. Global recombination in evolutionary strategy is a reproductive process, while migration in BBO is an adaptive process; it is used to modify existing habitats. As with other population-based optimization algorithms, some sort of elitism is typically incorporated in order to retain the best solutions in the population. This prevents the best solutions from being corrupted by immigration.

BBO has shown good performance both on benchmark problems and on real-world problems, including air-craft engine sensor selection [13], power system optimization [18, 19], groundwater detection [20], mechanical gear train design [21], and satellite image classification [22].

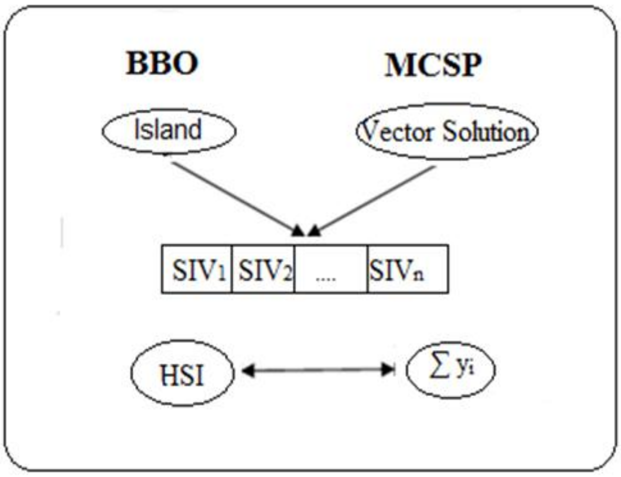

Figure. 1 Coding of solutions and the objective function

2.2 Solving MCSP with BBO

We recall the approach aims to determine the maximal number of species covered, while limiting the number of selected parcels to P. To use BBO, we must first find a good coding potential solutions and then find a fitness function to evaluate them.

A potential solution of the problem to be solved is a N-vector H, expressed in a form of binary sequence (with a value 1 if the parcel i is present in the set I and a value 0 otherwise). Consequently, in our case, the coding is straightforward: a habitat is composed with N SIVs. Each SIV value is 0 (corresponding parcel not covered) or 1 (parcel covered) and N is the total number of existing parcels. Each habitat in the archipelago corresponds to a particular N-vector solution H (Figure 1).

Further, as specified above, we have to search, among all the possible parcels subsets (of size P), for the one which presents the greatest number of species. This results in the maximization of the sum: $\sum Y_i$, where Y_i is a binary variable for species covering (it has 1 if species Y_i is covered, and has 0 otherwise).

As BBO algorithm is an optimum search algorithm, finding the maximum of a fitness function, we can easily conclude that in our case this function should be made equal to the sum ∑Yi. So we have:

$$F = \text{Max } \sum Y_i \quad (6)$$

However, this selective function ignores the fact we deal with a constrained problem, which implies that some habitats among the 2^N are not realistic. Solutions which do not satisfy the constraints will be penalized by setting their fitness to 0, migration rate μ to 0, and Immigration rate λ to1. As for the solutions satisfying the constraints, they will be sorted in decreasing order of the objective function.

In BBO algorithm, an initial population of several habitats (binary vector Y) is randomly generated. It evolves through a migration (emigration and immigration) and a mutation processes to reach an optimal solution.

During the evolution of the population, it is likely that the best solutions are modified, and therefore lost after the migration process and mutation. To avoid such a situation, an elitism operator is adopted. It copies the L best individuals in the new generation (the value of L has to be fixed by simulation).

The process of mutation and migration are BBO mechanisms used to derive a new solution Y based on the past solution, exploring the search space, trying to improve the best solution in terms of its fitness value (that is the total number of covered species). Changing an SIV in a Y vector by means of migration and mutation processes causes a solution to appear or disappear.

3 EXPERIMENTAL RESULTS

Tests were carried out on a 2.10 GHz Intel (R) Core (TM) 2 Duo CPU T6570 with 3GB memory. Data used is randomly generated. Each experiment is characterized by a triplet (P, Pm, L), where P is the population size, P_m the mutation parameter, L the elitism parameter and their values are fixed by simulation; All the results are averaged over 10 runs performed for each value of the triplet. We were first interested in the influence of each parameter, before fixing the parameters values.

Note that the execution time is stopped when there is no variation in the number of covered species.

Some experimental results are presented in Table 1, with:

- Number of parcels = 441 and Number of species = 426
- R: a randomly generated 441x426 matrix, Rij = number of species j in parcel i
- BBO parameter values : P_m = 0.005, L= 2, Generation Number = 200 and Population Size(P) = 100. (These parameters are obtained starting from the default values given in [13]).

Number of Parcels	Maximal umber of Species	Execution Time (sec
1	240	0.218
2	336	0.219
3	389	0.265
4	409	0.267
5	422	0.297
6	426	0.281

Table 1: Maximal Number of Species while limiting the number of Parcels

We observe that all the 426 species are covered with 6 parcels. Several other tests confirm the good behavior of our method. Furthermore, comparisons with an existing method using Harmony Search metaheuristic are performed.

3.1. BBO - based method vs. HS-based method

To make our work more self-contained, we first present below the Harmony Searh approach:

3.1.1 Harmony Search metaheuristic Approach

Harmony Search [14] was devised as a new metaheuristic algorithm, taking inspiration from the music improvisation process, where musicians improvise their instruments' pitches searching for a perfect state of harmony.

HS has been successfully applied to various discrete optimization problems such as tour routing [24], water network design [25], vehicle routing [26], and other industrial problems [27].

Analogies with optimization process are such that:

- Instrument i ↔ Decision variable x_i , i ϵ{1,2,…,n}
- Music note from instrument i ↔Value of variable x_i
- Harmony ↔ Solution vector
- Musical easthetic ↔ Fitness

HS algorithm mainly consists of the following:

Initialization of the Optimization Problem and Algorithm Parameters

The optimization problem is specified as follows:

Minimize (or Maximize) f(x)
subjected to xi ∈ Xi, i= 1, 2, . . .,n.

where f(.) is a scalar objective function to be optimized; x is a solution vector composed of decision variables xi, i= 1, 2, . . ., n; Xi is the set of possible range of values for each decision variable xi , that is, Lxi ≤ Xi ≤ Uxi, where Lxi and Uxi are the lower and upper bounds for each decision variable in the case of continuous decision variables and xi∈{xi(1), …, xi(k), …, xi(K)} when the decision variables are discrete. N is the number of decision variables.

In addition, the control parameters of HS are also specified in this step. These parameters are the Harmony Memory Size (HMS) i.e., the number of solution vectors (population members) in the HM (in each generation); the HM considering rate (HMCR); the pitch-adjusting rate (PAR) and the number of improvisations (NI) or stopping criterion.

Harmony Memory Initialization

In this step, each component of each solution vector in the parental population (initial Harmony Memory: HM) is randomly chosen. Then, obtained solutions are reordered in terms of the objective function value: $f(x^1) \leq f(x^2) \ldots \leq f(x^i) \ldots \leq f(x^{HMS})$, where $x^i = (x_1^i, x_2^i, \ldots, x_n^i)$ is the ith solution vector.

Harmony Memory is represented by the following matrix:

$$HM = \begin{bmatrix} x_1^1 & x_2^1 \ldots & \ldots & x_n^1 \\ x_1^2 & x_2^2 \ldots & \ldots & x_n^2 \\ \vdots & \ldots & \ldots & \ldots \\ x_1^{HMS} & x_2^{HMS} & \ldots & x_n^{HMS} \end{bmatrix}$$

New Harmony Improvisation

In this step, anew harmony vector $x' = (x_1', x_2', \ldots, x_n')$ is generated based on three rules: memory consideration; pitch adjustment and random selection. Generating a new harmony is called ‘improvisation’.

The memory consideration rule stipulates that the decision variable x_i, i= 1…n, takes a new value x_i' from HM matrix (in the set $\{x_i^1, x_i^2, \ldots, x_i^{HMS}\}$), with a probability HMCR (parameter value between 0 and 1), and takes a fresh value randomly selected from the set Xi with probability (1- HMCR). This rule can be summarized in equation (1).

$$x_i' \leftarrow \begin{cases} x_i' \in \left(x_i^1, x_i^2, \ldots, x_i^{HMS}\right) \text{with probability HMCR} \\ x_i' \in Xi \quad \text{with probability } (1 - \text{HMCR}) \end{cases} \tag{1}$$

Every component obtained in the memory consideration step is further examined to determine whether it should be pitch adjusted. This operation uses the parameter PAR (pitch adjustment rate) as follows:

- In the case of discrete variables: $x_i' = x_i(k)$

$$x_i' \leftarrow \begin{cases} xi(k + m) \quad \text{with probability PAR} \\ x_i' \quad \text{with probability } (1 - □\ \text{PAR}) \end{cases} \tag{2}$$

where : m∈ {-1, 1}.

- In the case of continuous variables:

$$x_i' \leftarrow \begin{cases} x_i' \ \text{rand}(0\ 1) * \text{bw with probability PAR} \\ x_i' \quad \text{with probability } (1 - \text{PAR}) \end{cases} \tag{3}$$

Where bw is an arbitrary distance bandwidth (a scalar number), and rand() is a uniformly distributed random number between 0 and 1. Evidently, the new harmony improvisation strategy is responsible for generating new potential variation in the algorithm and is comparable to mutation in standard evolutionary algorithms.

Harmony Memory update

If the new harmony vector $x' = (x_1', x_2', \ldots, x_N')$ is better than the worst harmony in the HM, judged in terms of the objective function value, the new harmony is included in the HM, and the existing worst harmony is excluded from the HM. This is actually the selection step ofthe algorithm where the objective function value is evaluated to determine if the new variation should beincluded in the population (HM).

Check Stopping Criterion

If the stopping criterion (maximum NI) is satisfied, the computation is terminated. Otherwise, New Harmony Improvisation and Harmony Memory update steps are repeated.

In similar manner than in section 2.2, we use HS algorithm to solve MCSP problem. Figure 2 gives a representation of coding of solutions and the objective function.

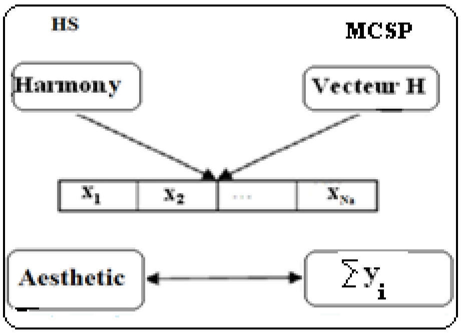

Figure. 2 Coding of solutions and the objective function

3.1.2 BBO - based method vs. HS-based method

Several tests are then performed on the same machine, using both approaches: BBO-based method and HS-based method.

Some experimental results (based on random data) are presented in Table 2, with:

- Number of parcels = 441 and Number of species = 426
- R: a randomly generated 441x426 matrix, Rij = number of species j in parcel i
- BBO parameter values : Pm = 0.005, Elitism = 2, Generation Number = 200 and Population = 100
- HS parameter values : PAR=0.4 ; Bw=1 ; HMCR=0.9

Number of parcels (P)	BBO	HS
1	236	233
2	352	343
3	386	386
4	411	413
5	422	422
6	**426**	424
7	426	426

Table 2: BBO - approach vs. HS – approach

We observe that all species are covered with 6 parcels when using BBO method, while 7 parcels are needed with HS.

4 CONCLUSIONS

The BBO method was applied to an ecological optimization problem (MCSP) where the number of preserved species in an area is to be maximized while limiting the number of considered area parcels. It was tested on a random generated data with 426 species and 441 parcels. The 426 species were covered by 6 parcels. Comparisons with HS – based method showed that the BBO algorithm produced better results.

REFERENCES

[1] Kirkpatrick, J. B., (1983) “An iterative method for establishing priorities for the selection of nature reserves: an example from Tasmania”, Biol. Conserv. 25, pp. 127-134.

[2] R.L. Pressey, R. L., (2002) “The first reserve selection algorithm : a retrospective on Jamie Kirkpatrick’s 1983 paper”, Prog. Phys. Geogr. 26(3), pp. 434-441.

[3] ReVelle, C., Williams, J. C., and Boland, J. J., (2002) "Counterpart Models in Facility Location Science and Reserve Selection Science", Environmental Modelling and Assessment, 7, pp.71-80.

[4] Toregas, C., Swain, R., ReVelle, C., Bergman, L., (1971), “The location of emergency service facilities”. Operations Research, 19, 1363-1373.

[5] Church, R., and ReVelle, C.S., (1974), “The maximal covering location problem”, Papers of the Regional Science Association, 32, 101-118.

[6] Possingham, H., Day, J., Goldfinch, M., Salzborn, F., (1993). The mathematics of designing a network of protected areas for conservation. In: Sutton, D., Cousins, E., Pierce, C. (Eds.), Decision Sciences, Tools for Today, Proceedings of the ASOR Conference, Adelaide, Australia. ASOR.

[7] Church,R. L., Stoms, D. M., and Davis, F. W., (1996) “Reserve selection as a maximal covering location problem”, Biol. Conserv. 76, PP. 105-112.

[8] Camm, J. D., Polasky, S., Solow, A., and Csuti,B., (1996) “A note on optimal algorithms for reserve site selection”, Biol. Conserv. 78, pp. 353-355.

[9] Csuti, B., S. Polasky, P. H. Williams, R. L. Pressey, J. D. Camm, M. Kershaw, A. R. Kiester, B. Downs, R. Hamilton, M. Huso, and K. Sahr, (1997) "A comparison of reserve selection algorithms using data on terrestrial vertebrates in Oregon," Biological Conservation, 80, pp. 83-97.

[10] Rosing, K. E., ReVelle, C. S., and Williams, J. C., (2002) "Maximizing Species Representation under Limited Resources: A New and Efficient Heuristic," Environmental Modeling and Assessment, 7(2), pp. 91-98.

[11] Mizumori, M. M., ReVelle, C. S., and Williams, J. C., (2005) "The maximal multiple-representation species problem solved using heuristic concentration", in The Next Wave in Computing, Optimization, and Decision Technologies, B. L. Golden, S. Raghavan, and E. A. Wasil, eds., Springer: New York, pp. 183-198.

[12] Camm, J. D., Norman, S.K., Polasky, S., Solow, A.R., (2002) “Nature reserve site selection to maximize expected species covered”, Operations Research; 50:946–55.

[13] Simon, D., (2008) “Biogeography-Based Optimization”. IEEE Trans. On Evol. Comput. Vol. 12, No. 6, pp 712-713.

[14] Geem, Z. W., Kim, J. H., Loganathan, G. V., (2001), “A new heuristic optimization algorithm: harmony search”, Simulation, Vol. 76, No. 2, pp. 60-68.

[15] Wallace, A., (2005) “The Geographical Distribution of Animals” (Two Volumes). Boston, MA: Adamant Media Corporation.

[16] Darwin, C., (1995), ”The Origin of Species”. New York: Gramercy.

[17] MacArthur, R., Wilson, E., (1967), “The Theory of iogeography”. Princeton, NJ: Princeton Univ. Press.

[18] Rarick, R., Simon, D., Villaseca, F., and Vyakaranam, B., (2009), “Biogeography-based optimiza-tion and the solution of the power flow problem”, IEEE Conference on Systems, Man, and Cybernetics, pp. 1029{1034.

[19] Roy, P., Ghoshal, S., and Thakur, S.,(2010), “Biogeography-based optimization for economic load dispatch problems”, Electric Power Components and Systems (38) pp. 166{181.

[20] Kundra, H., Kaur, A., and Panchal, V., (2009), “An integrated approach to biogeography based op-timization with case based reasoning for retrieving groundwater possibility”, 8th Annual Asian Conference and Exhibition on Geospatial Information, Technology and Applica-tions.

[21] Savsani, V., Rao, R., and Vakharia, D., (2009), Discrete optimisation of a gear train using bio-geography based optimisation technique, International Journal of Design Engineering (2) pp. 205-223.

[22] Panchal, V., Singh, P., Kaur, N., and H. Kundra., (2009), Biogeography based satellite image classification, International Journal of Computer Science and Information Security (6) pp. 269-274.

[23] Geem, Z. W., Williams, J. C., (2007) “Harmony Search and Ecological Optimization” International Journal of energy and Environment, Issue 2, Vol 1, pp.150-154.

[24] Geem, Z. W., Tseng, C. -L., and Park, Y., (2005), "Harmony Search for Generalized Orienteering Problem: Best Touring in China," Lecture Notes in Computer Science, 3612, 741-750.

[25] Geem, Z. W., (2006), "Optimal Cost Design of Water Distribution Networks using Harmony Search," Engineering Optimization, 38(3), 259-280.

[26] Geem, Z. W., Lee, K. S., and Park, Y., (2005),"Application of Harmony Search to Vehicle Routing," American Journal of Applied Sciences, 2(12), 1552-1557.

[27] Geem, Z. W., (2008), “Harmony Search Applications in Industry, (2008),” in Soft Computing Applications in Industry, Bhanu Prasad, Ed. Germany: Springer-Verlag, 2008, pp. 117-134.