Proceedings of the Symposium

2nd Swarm Intelligence Algorithms and Applications Symposium (SIAAS-09)

A symposium at the AISB 2009 Convention (6-9 April 2009)
Heriot-Watt University, Edinburgh, Scotland

Symposium Chairs
Prof. Aladdin Ayesh

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

ISBN - 1902956737

2nd Swarm Intelligence Algorithms and Applications Symposium (SIAAS-09)

A one-day symposium at AISB 2009 (6-9 April 2009).

PROGRAMME CHAIRS

Dr Aladdin Ayesh, De Montfort University, UK

INTRODUCTION

The increasing complexity of the current world can be observed each day. Sustainable development for example consists of economical and social systems management within natural environment. The understanding of the whole leads to what we call territorial intelligence. The way of modeling these complex systems is often based on interactive networks, dealing with the interconnection between all of the system components. Decision making on this complex world, need tools able to detect and manage emergent organizations through these networks. Distributed Artificial Intelligence (DAI) is the adapted conceptual trend which allows the proposal of some relevant solutions by relying on social and physical sciences models exhibited and observed in nature (e.g. ant colonies, molecular crystallization, etc.). In this search and management of emerging organization, swarm intelligence algorithms proved to be popular and effective methods to use. On the technological front, the increasing number of robotic systems, advances in nano technology, and the sheer complexity of modern enterprise systems, especially those boosting high degree of autonomy, makes the development of swarm intelligence timely and needed.

Swarm intelligence is one of timely topics in multi-agent and complex systems research with a wide range of applications. This is inline with AISB2009 themes reflected by the two topics of interest “Biologically- inspired and Socially-inspired Complex Systems”. The symposium will build on the strength of the program committee members’ knowledge and expertise in the area of swarm intelligence especially in relation to applications and engineering approach to swarm intelligence. It will also present a continuation from the AISB08 during which we held the same symposium successfully. The success of SIAAS-08 led to agreeing a special issue in Blackwell Computational Intelligence Journal. This year symposium has all the element of success with wide range of papers on new algorithms as well as innovative use of existing ones.

Table of Contents

A Long-term Swarm Intelligence Hedging Tool Applied to Electricity Markets

Filipe Azevedo[1], **Zita A. Vale**[1]

Abstract. This paper proposes a swarm intelligence long-term hedging tool to support electricity producers in competitive electricity markets. This tool investigates the long-term hedging opportunities available to electric power producers through the use of contracts with physical (spot and forward) and financial (options) settlement. To find the optimal portfolio the producer risk preference is stated by a utility function (U) expressing the trade-off between the expectation and the variance of the return. Variance estimation and the expected return are based on a forecasted scenario interval determined by a long-term price range forecast model, developed by the authors, whose explanation is outside the scope of this paper. The proposed tool makes use of Particle Swarm Optimization (PSO) and its performance has been evaluated by comparing it with a Genetic Algorithm (GA) based approach. To validate the risk management tool a case study, using real price historical data for mainland Spanish market, is presented to demonstrate the effectiveness of the proposed methodology.

1 INTRODUCTION

Long-term contractual decisions are the basis of an efficient risk management. On a vertical integrated electricity market, price variations were often minimal and heavily controlled by regulators. In this structure, electricity price evolution is directly dependent on the government's social and industrial policy, and price forecasting was mainly focused on the underlying costs (namely, fuel prices and technological innovation). Any price forecasting made on that basis was tended to be over the long-term. With electricity markets re-regulation and liberalization process, this changed dramatically. Ownership on this activity sector become private rather than public or a mixture of both and competitive markets, like pools or power exchanges, has been introduced for wholesale trading.

Due to the specific nature of the underlying asset, price forecast on liberalized electricity markets has been a hard task. Factors like charge characteristics (seasonality, mean-reversion and stochastic growth) and producer's characteristics (technology, generation availability, fuel prices, technical restrictions and import/export) are at the origin of the high price volatility in electricity markets. Trying to overcome this issue, several techniques have been used for short-term price forecast in electricity markets. In [1], artificial intelligent tools are applied to forecast spot prices, namely, a combination of neural networks and fuzzy logic are used to predict prices. In fact, besides the early scepticism, neural networks have now an extensive use in load [2] and in price [3, 4, 5] forecast. Fuzzy techniques together with neural networks are used to predict possible prices range [6, 7]. Stochastic processes are also used to analyze time series. In [8], ARIMA processes, a class of stochastic processes, were used to predict next-day electricity prices in mainland Spanish and in California markets. In [9], two forecasting tools based on dynamic regression and transfer function models are presented.

However, for the agents who want to maximize their profits and simultaneously to practice the hedge against the market price volatility, the use of forward, futures and options contracts become a constant in developed electricity markets. Those types of contracts have a maturity that goes from one year to several years in the future, turning more difficult the decision process related to contracts establishment if they aren't supported with a robust price forecast methodology.

Due to long delivery periods of the contracts described above, makes more sense to forecast the market price mean value for each month and continuously review the agent position (say once a month) or each time the agent needs to consider his contractual positions already locked, than forecast the market price for periods on an hour or half-hour basis for so long periods. It is difficult to find in the literature scientific documents that deal with this problem, which is a very important subject in electricity markets risk management with high market price volatility. However, it is not a good practice in risk management to take contractual decisions based exclusively on a single forecasted value. In [10] is presented a different approach for long-term price forecast. Making use of regression models, [10] has as main goal to find a maximum and a minimum monthly Market Clearing Price (MCP) average for a programming period, with a desired confidence level α. This methodology makes use of statistical information extracted from historical data. Due to the problem complexity, the parameters are obtained using the meta-heuristic Particle Swarm Optimization (PSO) [11, 12].

Finding an optimal portfolio for a market agent and in particular for the producers, which allow hedging against market price volatility and simultaneously increase their profits, is difficult due to the complexity of the optimization problem. The scientific literature reports some studies about this matter. In [13], solutions for electricity producers in the field of financial risk management for electric energy contract evaluation using efficient frontier as a tool to identify the preferred contract portfolio are proposed. A decision support system based on stochastic simulation, optimization and multi-criteria analysis is applied to electricity retailer in [14]. A statistical study of direct and cross hedging strategies using futures contracts in an electricity market is presented in [15, 16]. A framework to obtain the optimal bidding strategy of a thermal price-taker producer on a pool-based electric energy market is presented in [17].

[1] GECAD – Knowledge Engineering and Decision-Support Research Group of the Institute of Engineering – Polytechnic of Porto (ISEP/IPP), Rua Dr. António Bernardino de Almeida, 4200-072 Porto, Portugal. Email: {fta, zav}@isep.ipp.pt.

This long-term risk management tool makes use of a long-term price range forecast developed by the authors and presented in [10]. The proposed long-term risk management tool aims to find the "unknown optimal" portfolio in function of the risk aversion factor (λ) of the producer that maximizes the expected return and, simultaneously, allows the practice of the hedge against the market price volatility. To achieve this, the decision-support system maximizes a mean variance utility function (U) of the total return (π).

In this risk management tool was used a portfolio model based on utility functions instead of option pricing models [18, 19] because financial markets on electricity markets are incomplete (hedging instruments unavailable).

Uncertainties associated to generators availability, fuel prices, technical restrictions and weather conditions, turn difficult, if not impossible, to find a replicating portfolio that perfectly matches the future spot market payoffs. The power market exercise by some agents is also a source of uncertainty. In addition, several markets around the world are still on their child stage, with a small number of financial tools for an efficient risk management.

Another issue in power markets is that energy cannot be stored for later use. As a consequence, the strategy of buying the asset today to offset part of future losses does not apply. The closest strategy is to buy a forward or futures contracts. Based on that, the delivery price of these mentioned contracts should be equal to the expected spot market price for the delivery period, which not always happens. Consequently, we conclude that electricity markets are not complete, and so risk attitudes and mean-variance frontiers are still relevant.

Due to the complexity of the risk management tool, we make use of Particle Swarm Optimization (PSO) to find the optimal solution.

PSO performance has been evaluated by comparing it with a Genetic Algorithm (GA) [20, 21] to show that PSO is a very successful meta-heuristic technique for this particular problem.

The paper is organized as follows: in Section II, a short overview of PSO meta-heuristic is presented; in Section III contracts are presented and how their revenues are calculated; in Section IV the problem formulation of the risk management model is presented; in Section V a case study is presented and Section VI presents some relevant conclusions.

2 PARTICLE SWARM OPTIMIZATION

Particle swarm optimization [12, 15] is an evolutionary computational algorithm inspired on a natural system. On a given iteration, a set of solutions called "particles" move around the search space from one iteration to another accordingly to rules that depend on three factors: inertia (the particles tend to move in the direction they have previously moved), memory (the particles tend to move in the direction of the best solution found so far in their trajectory) and cooperation (the particles tend to move in the direction of the global best solution).

The movement rule of each particle can be expressed by

$$X_i^{new} = X_i + V_i^{new}$$

where,

X_i^{new}	represents the new position of the particle i
X_i	represents the current position of the particle i
V_i^{new}	represents the new velocity of the particle i

$$V_i^{new} = dec(t)\cdot V_i + rand_{i,k}\cdot\alpha_{i,k}\cdot(pbest_i - X_i) + rand_{i,j}\cdot\alpha_{i,j}\cdot .\left[pbest(gbest) - X_i\right]$$

where,

dec(t)	represents an inertia weight that decreases with the number of iterations
V_i	represents the previous velocity of the particle i
$rand_{i,k}$ $rand_{i,j}$	represents random weights acceleration, from a uniform distribution in [0,1], for each time step
$\alpha_{i,k}$	represents a weight fixed at the beginning of the process designated by cognitive acceleration parameter
$\alpha_{i,j}$	represents a weight fixed at the beginning of the process designated by social acceleration parameter
$pbest_i$	represents the particle i best position found so far
pbest(gbest)	represents the best global position of all particles found so far

The inertia term controls the exploration and exploitation of the search space. If the velocity is too high, then the particles could move beyond a global solution. On the contrary, if velocity is too low, the particles could be trapped into a local optimum. To achieve faster convergence and avoiding the problems described above, we make the inertia term vary with the number of iterations and limit the maximum velocity of particles to V_{max}.

3 CONTRACTS

Contractual diversification is the key issue for an efficient risk management. To achieve this, it is assumed that producers can make use of contracts with physical settlement (spot and forward contracts) and contracts with financial settlement (options contracts).

A. Spot Contracts

The spot market becomes the core of the main deregulated electricity markets around the world. Producers make extensive use of this market to sell their energy on an hour or half-hour basis. The revenue from the short position (who sells has a short position and who buys has a long position) obtained by the producer is dependent of the period i and scenario j and is given by:

$$r_{i,j}^{ss} = MCP_{i,j} \times e_i^{ss}$$

where,

$r_{i,j}^{ss}$	represents the revenue, in Eur, of the short position obtained by the producer in the spot market, for period i and scenario j
$MCP_{i,j}$	represents the Market Clearing Price, in Eur/MWh, for period i and scenario j
e_i^{ss}	represents the energy amount, in MWh, that the producer decides to sell in the spot market for period i

B. Forward Contracts

One of the most common methods used to hedge against spot price volatility is to establish forward contracts. Forward contracts are bilateral agreements in which two parts agree mutually on the characteristics (quantity, price, point of delivery and date/time). The payment is made only on a future date, eliminating the risk associated to price variation. Most of forward contracts are traded in organized and over-the counter (OTC) markets.

As stated previously, producers can make use of forward contracts to sell energy. So, the revenue from short forward positions obtained by the producer is given by:

$$r^{sf} = k^{sf} \times e^{sf}$$

where,

r^{sf}	represents the revenue, in Eur, of the short position obtained by the producer in forward contracts
k^{sf}	represents the delivery price, in Eur/MWh, of the forward contract
e^{sf}	represents the energy amount, in MWh, that the producer decides to sell in forward contracts.

In our method, the delivery period in forward contracts is the same of all period in analysis.

Because on forward contracts the delivery price is fixed, its revenue is only dependent on the delivery price and quantity established in the contract.

In this study, producers are not allowed to take any advantage of arbitrage opportunities, so not to obtain long forward positions.

C. Options Contracts

Traditionally, options in electricity markets have financial settlement. There is four positions types on options contracts and they are: short call, long call, short put and long put. However, in the decision-support system it is assumed that producers could only establish short call and long put positions. These positions are similar to the positions that the producer can establish to sell the produced energy with physical settlement. If the producer were allowed to establish the four positions types, the quantities to practice the hedge would be almost infinite if a financial limit is not established. In some electricity markets, options are on futures with daily settlement. The settlement price could be equal to the simple average of all 24 hours for Base Load Futures Contracts or equal to the simple average of the prices for the hours between 8:00 AM and 20:00 PM for Peak Load Futures Contracts. It is also assumed that they are European style options (European-style options can only be exercised at the beginning of the delivery date while American-style options can be exercised at any time until the delivery date).

The characteristics of electricity prices, such as mean reversion, high degree of skewness and non-constant volatility, exclude its modelling using commodity cost-of-carry models; Thus, Black & Sholes formula is not applicable to electricity option pricing. A procedure to evaluate the price of options in electricity markets, known as risk-neutral valuation, is presented in [16]. Binomial model could also be applied to evaluate electricity options price but it requires some adjustments.

For the short call position, the buyer only exercises the option if the *MCP* is greater than the exercise price. In our method, the delivery period in call options is the same of all period in analysis.

The payoff for the short call position is given by:

$$Payoff^{sc}_{i,j} = e^{sc} \times \left[\min(k^{sc} - MCP_{i,j}, 0) + p^{sc} \right]$$

where,

$Payoff^{sc}_{i,j}$	represents the payoff, in Eur, of the short call position, for the period i and scenario j
p^{sc}	represents the premium, in Eur/MWh, of the call option
k^{sc}	represents the delivery price, in Eur/MWh, of the call option
$MCP_{i,j}$	represents the Market Clearing Price, in Eur/MWh, for the period i and scenario j
e^{sc}	represents the energy, in MWh, associated to the short call position obtained by the producer.

Because the call option exercise is dependent on the system marginal price scenario, the short call position payoff is dependent on the scenario *j* considered for each period *i*.

For the long put position, the option buyer (producer) will exercise it if the *MCP* is lower than the exercise price.

The payoff for the long put position is given by:

$$Payoff^{lp}_{i,j} = e^{lp} \times \left[\max(k^{lp} - MCP_{i,j}, 0) - p^{lp} \right]$$

where,

$Payoff^{lp}_{i,j}$	represents the payoff, in Eur, of the long put position, for period *i* and scenario *j*
p^{lp}	represents the premium, in Eur/MWh, of the put option
k^{lp}	represents the delivery price, in Eur/MWh, of the put option
$MCP_{i,j}$	represents the Market Clearing Price, in Eur/MWh, for period *i* and scenario *j*
e^{lp}	represents the energy, in MWh, associated to the long put position obtained by the producer.

From the last equation it is clear that the long put position payoff is positive only if the *MCP* is higher than the exercise price.

4 OPTIMIZATION PROBLEM

To find optimal energy quantities establishing on each contract type, it was developed an optimization problem based on a mean-variance of the return. This formulation allows finding the optimal energy quantities that maximizes the profits and simultaneously practices the hedge against the *MCP* volatility in function of the producer risk aversion factor.

The mathematical formulation is stated as follows:

Maximize $\quad U(\pi) = E(\pi) - \lambda \times Var(\pi)$

Subj. to:

$$e_{min} \le e^{cs}_i + e^{cf} \le e_{max}$$

$$e^{cs}_i, e^{cf}, e^{cc}, e^{lp} \ge 0$$

where,

$$E(\pi) = E\left(\pi^{máx}\right) + E\left(\pi^{min}\right)$$

and,

$$Var(\pi) = \sum_{i=1}^{2}\sum_{j=1}^{2} cov_{i,j}\left(\pi^{máx}, \pi^{min}\right)$$

with,

$$\pi^{máx} = \left[\pi_1^{máx}, \ldots, \pi_T^{máx}\right]$$

and,

$$\pi^{min} = \left[\pi_1^{min}, \ldots, \pi_T^{min}\right]$$

where,

π	represents the producer return, in Eur, for the entire period in analysis
$E(\pi)$	represents the expected value of the return, in Eur, based on the forecasted price interval for the entire period in analysis
$Var(\pi)$	represents the variance of the return, in Eur, based on the forecasted price interval for the entire period in analysis
$\mathrm{cov}_{i,j}\left(\pi^{max}, \pi^{min}\right)$	represents the element *(i,j)*, in Eur, of the covariance matrix of the returns for all periods *i* based on maximum and minimum price forecast
π_i^{max}	represents the period *i* return, in Eur, based on the maximum price forecast
π_i^{min}	represents the period *i* return, in Eur, based on the minimum price forecast
T	represents the number of the considered periods for the entire period in analysis
λ	represents the producer risk aversion factor
e_{min}	represents the minimum energy, in MWh, that the producer can produce
e_{max}	represents the maximum energy, in MWh, that the producer can produce
e_i^{ss}	represents the energy amount, in MWh, that the producer decides to sell on the spot market for period *i*
e^{sf}	represents the energy amount, in MWh, that the producer decides to sell in forward contracts
e^{sc}	represents the energy, in MWh, associated to the short call position obtained by the producer
e^{lp}	represents the energy, in MWh, associated to the long put position obtained by the producer.

The mean-variance formulation resemble closely the Value-at-Risk (*VaR*) formulation and have as main advantage to be computationally more efficient for a given risk aversion factor λ. Moreover, *VaR* formulation needs higher order of information about the joint probability distribution of the payoffs and is highly sensitive to the high impact of low probability events, which create "fat tails" in payoff distribution. In this formulation we assumed the risk aversion factor λ is equal for the whole period in analysis.

The return π for each period *i*, expressed in Eur, is a function of the considered minimum or maximum price forecast scenario *j* for that period, and is equal to the sum of all revenues and options payoffs minus the costs of production.

Mathematically, the return π is given by:

$$\pi_{i,j} = r_{i,j}^{ss} + r^{sf} + P_{i,j}^{sc} + P_{i,j}^{lp} - C_{i,j}$$

with,

$$C_{i,j} = C(e_i^{ss} + e^{sf})$$

Options contracts have financial settlement; the total production cost is only dependent on the energy that the producer will sell on spot market, and on forward contracts, meaning that is only dependent of the energy established on contracts with physical settlement.

A. *Penalty functions*

Due to optimization problem complexity, PSO was used to find the optimal solution and results were compared with GA results.

To satisfy constraint the first restriction of the optimization problem for each period *i*, was added the following penalty function:

$$p_{f1} = \begin{cases} 0 \quad if \quad e \geq e_{min} \; and \; e \leq e_{max} \\ e^{100\times d^2} - 1 \quad otherwise \end{cases}$$

where,

$$d = min\left[\left|e - e_{min}\right|, \left|e - e_{max}\right|\right]$$

To guaranty that all variables are positives, was added the following penalty function:

$$p_{f2} = \begin{cases} 0 & if \quad e_i^{ss,sf,sc,lp} \geq 0 \\ e^{100\times e^2} - 1 & otherwise \end{cases}$$

where,

$$e = \left|e_i^{ss,sf,sc,lp}\right|$$

B. *PSO and GA Parameters*

The main parameters of PSO and GA, used finding the best solution are presented in table 1 and table 2, respectively.

Besides these parameters being dependent on the fitness function, experimentations show that the number of evaluations used does not compromise the results.

Nº. of particles	20
Nº. of iterations	20000
Nº. of evaluations	400000
Cognitive acceleration	2
Social acceleration	2
Initial inertia weight	0.9
Final inertia weight	0.2
Maximum velocity (V_{max})	0.1

Table 1. PSO Parameters

Population size	50
Nº. of generations	8000
Nº. of evaluations	400000
Crossover rate	0.8
Mutation rate	0.2

Table 2. GA Parameters

C. Producer Characteristics

It was assumed that producer cost function is equal for the entire period in analysis (one year) and is given by:

$$C(P_g) = 100 + 0.3 \times P_g + 0.02 \times P_g^2$$

where.

P_g in MW, C in Eur/h, P_g^{max} = 200 MW and P_g^{min} =5 MW.

The cost of sales (like taxes, market commissions and others) is not addressed. Moreover, there is just as much risk in the cost of sales as there is in the generation of revenue.

D. Contracts Characteristics

Options contracts characteristics with delivery period for the year 2007 are presented in Table 3.

	Exercise Price (Eur/MWh)	Premium (Eur/MWh)
Short Call	42.00	2.50
Long Put	45.00	5.00

Table 3. Options Contracts Characteristics

It was assumed that forward contracts with delivery period for the year 2007 are traded at a price equal to 40 Eur/MWh.

5 CASE STUDY

In this case a producer aims (in December 2006) to find the optimal contracts portfolio for the entire year of 2007. However, although to be beyond the purpose of this work, the producer must adjust its contractual positions continuously (say once a month) and whenever he needs to reconsider his contractual positions already established in forward and other contracts, before adjusting the portfolio.

Using the method presented in [10], that also makes use of PSO, the monthly price range average forecast for the year 2007 is shown on figure 1.

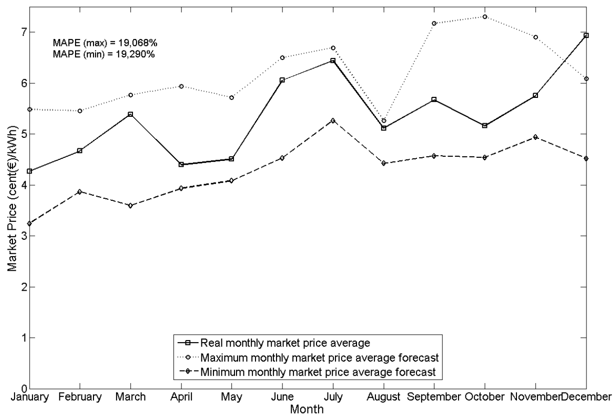

Figure 1. Monthly Market Price Range Forecast for the Year 2007 in Mainland Spanish Market, with Confidence Level of α=95%

An evaluation of PSO and GA performance for this particular problem has been carried out. The algorithms' stopping criterion was the maximum number of evaluations (fixed in 400,000 evaluations). With 20 particles in the PSO 20,000 iterations were performed. For GA a population size of 50 individuals and 8,000 generations was used. Due to random initialization, the trajectory for each run is different; so, we used 10 runs to calculate the average and the standard deviation of the results.

Due to the problem complexity, the entire period was divided in sub-periods of one month of duration allowing reducing the number of variables and, consequently, turning the optimization problem lighter.

As results, table 4 and table 5 present the average quantities, in MWh, for each contractual position and risk aversion factor, using PSO and GA, respectively.

Position	Average Quantity (MWh)			
	λ=0	**λ=1**	**λ=2**	**λ=3**
Short Spot	2.1×10^6	1.2×10^6	9.9×10^5	7.7×10^5
Short Forward	2.7×10^3	6.4×10^5	5.9×10^5	3.8×10^5
Short Call	0.839	1.3×10^6	1.4×10^6	5.4×10^5
Long Put	0.250	6.9×10^5	1.3×10^6	7.2×10^5

Table 4. Average Quantities, in MWh, to Establish by Contractual Position and Risk Aversion Factor using PSO

Position	Average Quantity (MWh)			
	λ=0	**λ=1**	**λ=2**	**λ=3**
Short Spot	1.8×10^6	1.1×10^6	1.3×10^6	1.1×10^6
Short Forward	2.4×10^5	5.4×10^5	4.4×10^5	4.8×10^5
Short Call	145.017	8.7×10^5	1.4×10^6	1.0×10^7
Long Put	444.929	9.8×10^5	7.8×10^5	4.1×10^5

Table 5. Average Quantities, in MWh, to Establish by Contractual Position and Risk Aversion Factor using GA

The results standard deviation using PSO and GA is presented in table 6 and table 7, respectively.

Position	Quantities Std. Deviation (MWh)			
	λ=0	**λ=1**	**λ=2**	**λ=3**
Short Spot	0.004	2.026	112.788	24.277
Short Forward	1.7×10^{-4}	1.028	6.233	0.979
Short Call	6.1×10^{-6}	26.243	7.797	30.443
Long Put	8.6×10^{-6}	45.682	75.483	5.041

Table 6. Quantities Std. Deviation, in MWh, to Establish by Contractual Position and Risk Aversion Factor using PSO

Position	Quantities Std. Deviation (MWh)			
	λ=0	**λ=1**	**λ=2**	**λ=3**
Short Spot	213.693	6.267	227.004	321.845
Short Forward	17.807	1.534	2.237	5.292
Short Call	0.0079	68.864	29.499	159.678
Long Put	0.0229	145.471	94.215	9.6719

Table 7. Quantities Std. Deviation, in MWh, to Establish by Contractual Position and Risk Aversion Factor using GA

Comparing the standard deviation for each solution (table 6 and table 7), we conclude that PSO is more robust than the GA.

The mean and the standard deviation of the fitness functions for the 10 runs and for each risk aversion factor are presented in table 8. Table 8 also includes the mean time necessary to reach the optimal solution for PSO and GA.

It can be verified from table 8 that, for this particular problem, PSO is faster than GA (smaller mean time), finds better solutions (smaller mean fitness value) and is more robust (smaller standard deviation). These simulations were made on an ASUS L5GX laptop, P4 3.2 GHz processor and 1 GB of memory.

Algorithm	Mean Fitness Value	Std. Fitness Value	Mean Time (sec.)
PSO (λ=0)	1.3639×10^7	10.9801	113.1464
GA (λ=0)	1.3181×10^7	2.9971×10^5	858.5784
PSO (λ=1)	9.5269×10^6	1.8706×10^5	107.2944
GA (λ=1)	7.8527×10^6	7.4777×10^5	885.0692
PSO (λ=2)	7.5300×10^6	2.6324×10^5	107.4961
GA (λ=2)	1.8101×10^6	2.3825×10^6	880.0213
PSO (λ=3)	6.3729×10^6	2.7286×10^5	106.3816
GA (λ=3)	4.6687×10^6	4.5523×10^5	868.8086

Table 8. PSO and GA Fitness Function Comparison

Because PSO achieve better results in this particular problem, in figure 2 and figure 3 is presented its results for the expected return and the associated risk for each month, as function of the risk aversion factor λ, respectively.

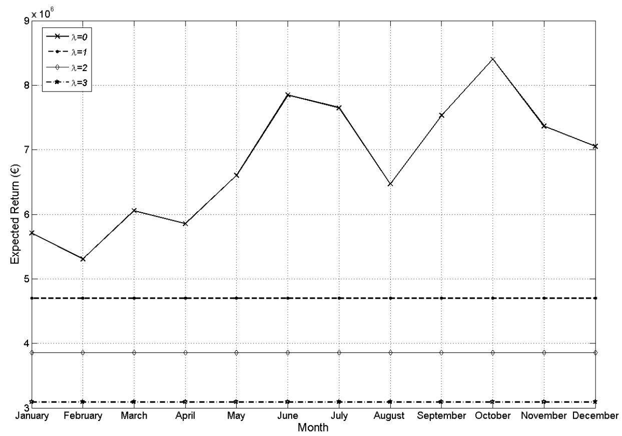

Figure 2. Producer Expected Return in Function of Risk Aversion Factor λ

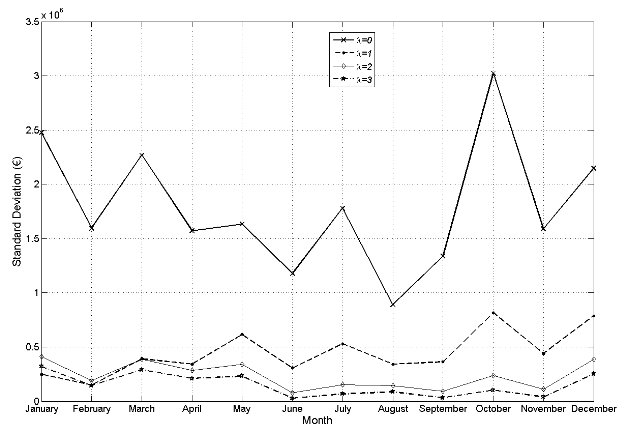

Figure 3. Risk in Function of the Risk Aversion Factor λ

From figure 2 and figure 3 we conclude that, for the same risk aversion factor λ, the bigger the expected return the bigger the risk (standard deviation of the return) that the producer is exposed to. Analyzing figure 2 and figure 3 we verify that the risk (standard deviation of the return) is inversely proportional to the risk aversion factor λ, and so is the energy that the producer will sell in the spot market. This happens because the lower the risk aversion factor the most indifferent the producer will be to the risk and therefore he will have more risky attitudes and sell more energy on the spot market, as it can be seen in figure 4.

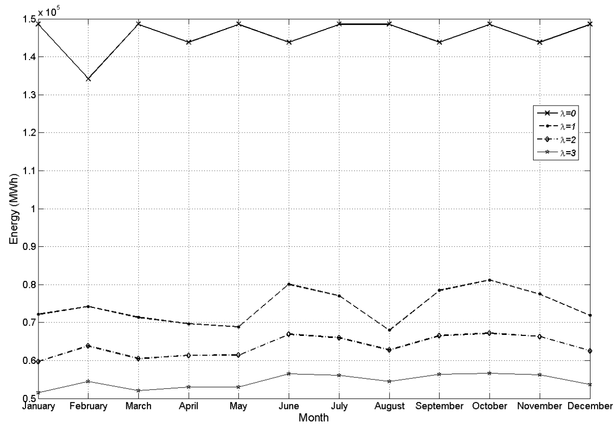

Figure 4. Optimal Energy Quantities that Producer Should Sell in Spot Market in Function of Risk Aversion Factor λ

6 CONCLUSIONS

With electricity markets liberalization, long-term contractual decisions are more difficult on an efficient risk management.

This paper proposed a new long-term risk management tool, which allows maximizing the producers expected return while practicing the hedge against spot price volatility based on the risk aversion factor.

Due to the optimization problem complexity, a Particle Swarm Optimization (PSO) meta-heuristic technique has been used. Its performance has been evaluated by its comparison with a Genetic Algorithm (GA) based approach. Actually the authors work in the application of Ant Colony System (ACS) Algorithm to solve the optimization problem with the aim to compare its results with the PSO performance, and the comparison will be reported shortly.

However, every risk management tools needs an efficient price forecast methodology. Trying to give an answer to that need a regressive model was used which enables the electricity market agents to forecast the monthly market price average range up to one year into the future. This model may find the price range value, with a certain confidence level, based on historical statistical data. The main advantage of this method is the fact that it does not make any statistical assumption relating to the market price distribution function.

Based on the results, it was proven that Particle Swarm Optimization (PSO) has significant advantages compared with GA in terms of robustness and computation time based in simulation results.

REFERENCES

[1] C. P. Rodriguez and G. J. Anders, "Energy Price Forecasting in the Ontario Competitive Power System Market", *IEEE Transactions on Power Systems*, vol. 19, no. 1, February 2004.

[2] H. Chen, C. Cañizares and A. Singh, "ANN-based Short-Term Load Forecasting in Electricity Markets", *Power engineering Society Winter Meeting*, IEEE, vol. 2, February 2001.

[3] B. R. Szkuta, L. A. Sanabria and T. S. Dillon, "Electricity Price Short-Term Forecasting Using Artificial Neural Networks", *IEEE Transactions on Power Systems*, vol. 14, no. 3, August 1999.

[4] F. Azevedo and Z. A. Vale, "Short-term Price Forecast from Risk Management Point of View", *13th International Conference on Intelligent Systems Application to Power Systems – ISAP 2005*, Arlington – EUA, November 2005.

[5] F. Azevedo e Z. A. Vale, "Forecasting Electricity Prices with Historical Statistical Information using Neural Networks and Clustering Techniques", *PSCE – Power Systems Conference and Exposition*, Georgia Atlanta –USA, October 2006.

[6] T. Niimura, H. Ko and K. Ozawa, "A Day-Ahead Electricity Price Prediction Based on a Fuzzy-Neuro Autoregressive Model in a Deregulated Electricity Market", *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 2, pp. 12-17, May 2002.

[7] L. Hongjie, W. Xiugeng, Z. Weicun and X. Guohua, "Market Clearing Price Forecasting Based on Dynamic Fuzzy System", *Proceedings of the 2002 International Conference on Power System Technology*, vol. 2, pp. 13-17, October 2002.

[8] J. Contreras, R. Espínola, F. J. Nogales and A. J. Conejo, "ARIMA Models to Predict Next-Day Electricity Prices", *IEEE Transactions on Power Systems*, vol. 18, no. 3, February 2003.

[9] F. J. Nogales, J. Contreras, A. J. Conejo and R. Espínola, "Forecasting Next-Day Electricity Prices by Time Series Models", *IEEE Transactions on Power Systems*, vol. 17, no. 2, May 2002.

[10] F. Azevedo, Z. A. Vale e P. B. Moura Oliveira, "Long-term Price Range Forecast Applied to Risk Management Using Regression Models," *Proceedings ISAP 2007 - 14th International Conference on Intelligent Systems Application to Power Systems*, Kaohsiung - Taiwan, November 2007.

[11] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", *IEEE International Conference on Neural Networks*, Perth, Australia, 1995.

[12] Y. Shi and R. Eberhart, "Parameter Selection in Particle Swarm Optimization", *Proceedings of the Seventh Annual Conference on Evolutionary Programming*, pp. 591-601, 1998.

[13] R. Bjorgan, C. Liu and J. Lawarrée, "Financial Risk Management on a Competitive Electricity Market", *IEEE Transactions on Power Systems*, vol. 14, no. 4, November 1999.

[14] S. Makkonen, R. Lahdelma, A. Asell and A. Jokinen, "Multi-criteria Decision Support in the liberalized Energy Market", *Journal of Multi-Criteria Decision Analysis*, Anal. 12:27 – 42, 2003.

[15] E. Tanlapco, J. Lawarrée, and C. Liu, "Hedging With Futures Contracts in a Deregulated Electricity Industry", *IEEE Transactions on Power Systems*, vol. 17, no. 3, August 2002.

[16] R. Bjorgan, H. Song, C. Liu and R. Dahlgren, "Pricing Flexible Electricity Contracts", *IEEE Transactions on Power Systems*, vol. 15, no. 2, May 2000.

[17] J. Conejo, F. J. Nogales and J. M. Arroyo, "Price-Taker Bidding Strategy Under price Uncertainty", *IEEE Transactions on Power Systems*, vol. 17, no. 4, November 2002.

[18] F. Black and M. Scholes, "The Pricing of Options and Corporate Liabilities", *Jornal of Political Economy*, no. 81, pp. 637 – 659, June 1973.

[19] R. C. Merton, "Theory of Rational Option Pricing", *Bell Journal of Economics and Management Science*, no. 4, pp. 141 – 183, 1973.

[20] J. Holland, "Genetic algorithms," *Scientific American*, pp. 66 – 72, July de 1992.

[21] M. Mitchell, "An Introduction to Genetic Algorithms," *MIT Press*, 1996.

Intermediate selection pressure bring in the emergence of altruism and common words

Yoshimitsu Hashizume † and **Jun Nishii** †† [1]

Abstract. Many animals show altruistic behavior by using signals, such as alarm calls. However, few study have discussed the coevolution of altruistic behavior and common words among agents, although many studies have discussed how altruism have evolved. In this study we performed simulation experiments in order to explore the condition of the emergence of altruistic behavior with common words between agents who pursue their own profit. The result suggests that some kinds of altruism and common words are emerged under intermediate selection pressure and altruism can be a dominant strategy for agents under such a condition. Our results also suggest that the dynamics of resources would be one of important keys to the understanding of the emergence of altruism.

1 Introduction

According to Richard Dawkins (1989), as a result of natural selection, selfish genes that are capable of reproducing themselves accurately and efficiently have survived [1]. On the other hand, individuals that are vehicles of the genes seem to show not only selfish behavior but also altruistic behavior, and to have developed common vocabularies not only to profit together but also to bring benefit to others by informing the existence of foods and hazards. Why do individuals that should take selfish behavior for the preservation of their genes have such common vocabularies and take altruistic behavior?

W. D. Hamilton proposed the idea that the basic unit for the evolution is a fragment of a gene and proposed the idea of kin selection [2, 3]. On the other hand, D. S. Wilson stated between-group selection is the basic key of the evolution of altruism and proposed the multi-level selection [4]. Even in recent years there are many arguments between these ideas [5, 6]. Then, is the concept of between-group selection necessary to the emergence of altruistic behavior?

The mechanism of the emergence of altruistic behaviors has been discussed without considering communication ability explicitly in most studies. However, a kind of communication would be necessary for cooperative behaviors. Many animals behave altruistic by using signals, for examples vervet monkeys warn the existence of predators by alarm call. Therefore, the emergence of signals commonly used among agents would be an important factor for the emergence of altruistic behavior. Some simulation studies have been done to discuss the evolution of common words [7, 8]. However, in most of the studies a word (or a symbol) to express an object is assumed to be acquired among agents by a mutual mimicry of words for an object through common experiences, which implies that a learning mechanism of common words is prepared a priori in these studies. However, the emergence of common words should be explained from a view point of natural selection.

We have examined the condition of the emergence of cooperative behavior and common words among agents who pursue their own profit by computer simulations [9]. The simulation results indicated that the words which point out an object and a cooperative relation emerge when agents can obtain more profit by the cooperation. However, neither a word nor a cooperative relation for non-profitable objects such as a predator was emerged. In the previous study, the duplication and death of agents were determined by a comparative assessment, that is, agents who obtained higher score than others could survive. It would be a reason why altruistic behavior did not emerge, because if one informs others hazard, its own relative domination falls. Then, does the factor of absolute assessment for selection bring altruistic behavior among agents?

In this paper we report a result of simulation experiments to consider the mechanism of the coevolution of altruistic behavior and common words.

2 Simulation method

We prepared a virtual field on a computer where foods and predators are put on, and agents move around. Each virtual agent determines its action, i.e. the direction of movement and the word to emit, from the state of the current cell and a word from a neighbor. The input-output relation is improved through evolutional process. We examined the condition of the emergence of common words to express objects and cooperation among agents. Here we call signals which agents transmit to others as words. Details of the simulation method are as follows.

2.1 Virtual field

We prepared a two dimensional torus composed of 10×10 cells as a virtual field. Foods and predators which are objects giving positive and negative rewards for agents, respectively, are put on each cell with the probability of 0.5% and 20%, respectively, as an initial condition, and added to empty cells with the probability of 1% and 0.5%, respectively, for every simulation step. In order to prevent extinction of agents, the appearance probability of foods was set higher than that of predator. We also have confirmed that the following results do not change qualitatively by the change of these values. If an agent stays at a cell with a predator for three consecutive steps, the predator disappears. If an agent stays five steps at a cell with a food, the food disappears.

[1] Graduate School of Science and Engineering, Yamaguchi University, email: {zenko†, nishii††}@bcl.sci.yamaguchi-u.ac.jp

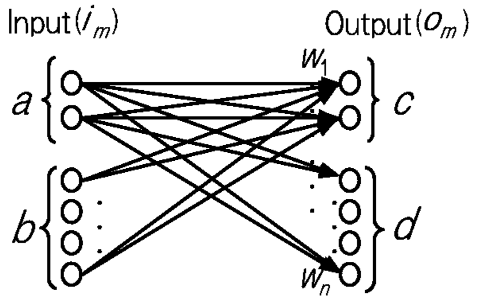

Figure 1. The input-output relation of an agent. The output shows the direction of a movement in the next step (signal c) and a word to speak to neighbors (signal d) are determined by the weighted sum of the elements of input signal. The input signal shows the information on the current position (signal a), and the word transmitted from a neighbor (signal b). For the simplification of data analysis, no connection from the received word (signal b) to the word to emit (signal d).

2.2 Agents

In each simulation step, each agent decides its movement and a word to speak to neighbors from input signals. The input signal involves two kinds of information. One is a 2-bit binary and shows the information of the current position (signal a in Fig.1) : the existence of a food (01), a predator (10), or nothing (00). The other is a 4-bit binary which shows a word transmitted from another agent in its eight neighboring cells (signal b in Fig.1) for simplicity. When an agent receives more than one word at the same time, the agent chooses one at random. When no word is received or no agent exists in neighboring cells, the signal was set as 0000. The output signal is determined by the weighted sum of the elements of the input signal, where the weights (w_i in Fig.1) take binary values. When the element of the output signal takes a non-binary value, it is replaced by the remainder of the value divided by 2. The output signal shows a word to emit (signal d in Fig.1) and the action taken at the next step (signal c in Fig.1) : moving to the cell where the received word was emitted (01), moving to a cell where no word was emitted (10), staying at the current cell (11), or moving to its nine neighboring cells including the current position, randomly (00). In the following sentences, we define 'the approaching action' as 'moving to the cell where the received word was emitted' and 'the avoiding action' as 'moving to a cell where no word was emitted or staying at the current cell'. As an initial condition the number of agents on the virtual field was set as 20, and the weights were set randomly. The weight value of each agent is fixed in its life time.

2.3 Survival conditions and Learning method

Each agent consumes 0.1 point at every simulation step as metabolic cost, and 0.5 points when they emit words. If an agent stays in a cell with a food, it obtains a reward of 1 point, and if an agent is in a cell with a predator, it obtains a negative reward of y points, where the reward was changed as $y = -1, -2, \cdots, -10$ for each simulation condition.

The agent whose point reaches 100 points makes an offspring. The weight values which determine the input-output relation are copied from the parent to the offspring, but copy error occurs by a given mutation rate. The agent whose point becomes below a given threshold x dies and the threshold is changed as $x = -30, -60, -90, \cdots, -900$ for each simulation condition, where the highest threshold $x = -30$ means the most severe environment for agents to survive. Therefore, the threshold can be regarded as a kind of selection pressure for the survival of agents. One simulation involves 100,000 steps and 1,000 simulations were performed for each reward condition y and threshold x from random initial states and random weight values. The obtained results, such as the number of agents which take a specific action, were averaged over a simulation steps and were normalized by the average of the number of agents in the simulation, and then averaged again over 1000 simulations for each simulation condition of x and y.

3 Results and discussions

In the following sentences, we define 'a word expressing a food (or a predator)' as 'a word emitted by an agent in a cell with a food (or a predator)' and 'approaching (or avoiding) action for "the word food" (or "the word predator")' as 'the action of approaching (or avoiding) the cell where a word expressing a food (or a predator) is emitted'.

3.1 Coevolution of altruistic behavior and common words

Figure 2, 3, 4 and 5 show the maximum duration steps in which agents successively choose an action, approaching or avoiding, when agents receive words expressing a food and a predator, respectively, against the threshold x of death condition and reward y of a predator. Figure 2 and 5 show that approaching action for "the word predator" and avoiding action for "the word food" were not chosen by agents. In other words, agents who chose such action were weeded out.

Figure 3 and 4 show that maximum duration of the avoiding action for "the word predator" and the approaching action for "the word food" is observed in the area which gives intermediate selection pressure, i.e. along the diagonal line from the upper right to the lower, respectively. In the upper left and lower right areas of Fig. 3 and 4 agents do not take the avoiding action for "the word predator" and the approaching action for "the word food" over a long duration, and agents tend to emit no words. Because in the upper left area the threshold x is low and the damage y by a predator is small, which makes agents scarcely die, therefore, agents do not need to speak for searching foods. In the lower right area threshold x is high and damage y is large, which would prevent agents to lose points by emitting words. However, except such areas the avoiding action for "the word predator" stably emerged in almost every environment (Fig. 3). These results suggest that the cooperative behavior to inform the existence of the foods and predators stably emerges under intermediate selection pressure with absolute evaluation in spite that such altruistic behavior can decrease the relative fitness of doers.

Figure 6 and 7 are the example of transition of words expressing a food and a predator, respectively, when the reward of predator is -7 points and dead threshold is -150 points. In each figure, the solid line shows the number of agents which did not emit words for a food. However, it is not invisible in the figures because the line is on the horizontal axis. The other lines show the number of agents which emitted certain word to express an object, respectively. These figures show that common words expressing a food and a predator emerged and stably observed, although, transition word expressing a predator was sometimes observed as in Fig. 7. Such emergence of common words was observed under intermediate selection pressure where both types of cooperative behaviors, avoiding predators and approaching foods by informing the existence each other, were

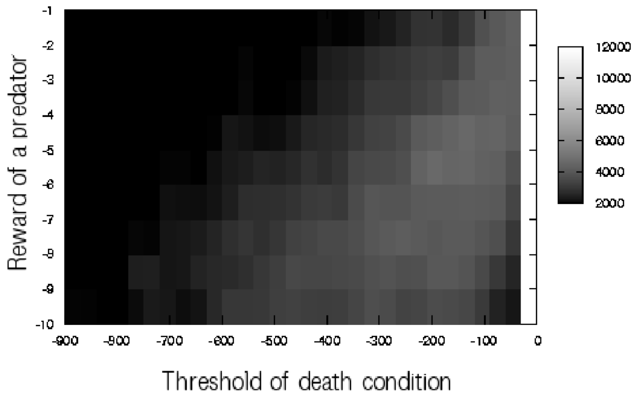

Figure 2. The average of maximum duration of the approaching action for "the word predator". The abscissa and ordinate are the threshold x of death condition and the reward condition y, respectively. The average was taken over 1000 simulations for each condition.

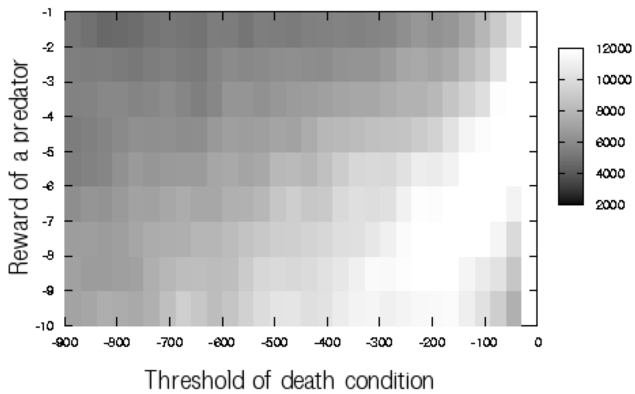

Figure 3. The average of maximum duration of the avoiding action for "the word predator". The abscissa and ordinate are the same as Fig. 2.

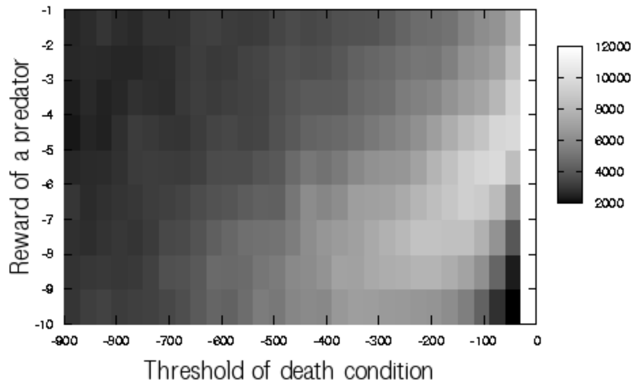

Figure 4. The average of maximum duration of the approaching action for "the word food". The abscissa and ordinate are the same as Fig. 2.

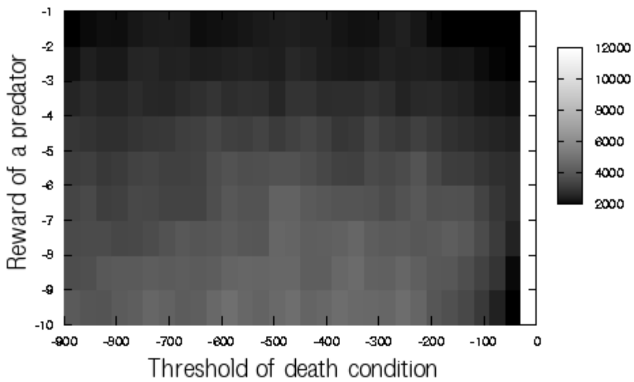

Figure 5. The average of maximum duration of the avoiding action for "the word food". The abscissa and ordinate are the same as Fig. 2.

emerged. Figure 8 and 9 show the action chosen by agents which received "the word food" and "the word predator", respectively. The solid lines show the ratios of the action "going to the cell where words expressing a food" and "going to the cell where words expressing a predator", respectively. The dashed lines show ratios of the actions "avoiding the cell where words expressing a food" and "avoiding the cell where words expressing a predator", respectively. It is clear that most agents go to the cell where words expressing a food is emitted (Fig. 8) and avoid the cells where words expressing a predator is emitted (Fig. 9). Therefore, both of common words and altruistic behaviors emerged under the intermediate selection pressure. However, in other area, i.e. , in the upper left area or in the lower right area, agents tend to emit no words expressing either a food or a predator. For example, Fig. 10 and 11 show an example of transition of words expressing a food and a predator, respectively under the condition that the reward of predator y is -1 point and the death threshold x is -900 points. The meaning of the lines is the same as those in Fig. 6 and 7. These graphs show that most agents emit no words for objects. Figure 12 and 13 show the action chosen by agents which received "the word food" and "the word predator", respectively. No cooperative behaviors emerged in these conditions (Fig. 12 and 13). Therefore, the coevolution of altruistic behavior and common words was confirmed only under the intermediate selection pressure.

3.2 Effect of communication

We examined how the communication affects the number of agents and objects. Figure 14, 15, and 16 show the mean number of agents, foods and predators, respectively, against the threshold x of death condition and reward y of a predator. Figure 17, 18, and 19 show the same ones as Fig. 14, 15, and 16, respectively, except that the simulation was done under the condition that agent always emit no word.

Figure 14 and Fig. 17 show that the number of agents who have communication ability is greater than the number of agents who do

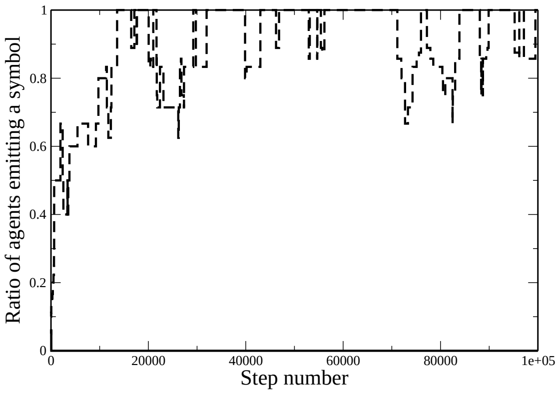

Figure 6. A typical pattern of transition of words expressing a food. The abscissa is step number. The ordinate is the ratio of agents emitting each symbols to the number of agents. The solid line shows the number of agents which did not emit words for a food. The dashed line shows the number of agents which emitted '1000' to express a food. In this simulation reward of predator is -7 points and dead threshold is -150 points.

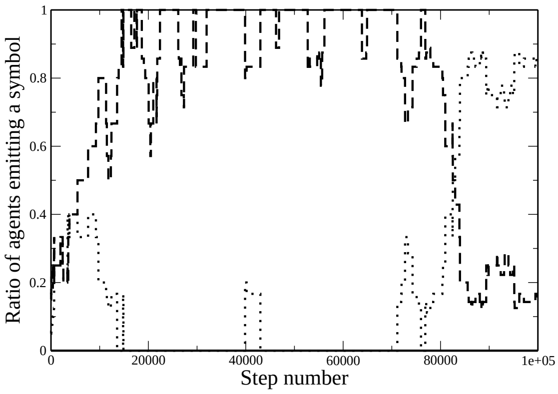

Figure 7. A typical pattern of transition of words expressing a predator. The abscissa and ordinate are the same as Fig. 6. The solid line shows the number of agents which did not emit words for a predator. The dashed and dotted line shows the number of agents which emitted '1101' to express a predator and '1110', respectively. The simulation condition is the same as Fig. 6.

not have one. Interestingly, in spite of the change of the number of agents, the number of objects shows little change by the communication (Fig. 15, 16, 18, and 19). These results show that some kinds of cooperation can exploit the dynamics of objects and increase the carrying capability of agents.

4 Conclusion

In this study we examined the condition under which altruistic behavior and common words emerge among agents that pursue their own profit. The results suggest that altruistic behaviors such that informing the existence of objects to others and the common word easily emerges among agents under the selection by absolute evaluation in spite that telling the existence of predator to others does not bring benefit to the speaker but might decrease its relative dominance to others.

D. S. Wilson said "Altruism decreases the relative fitness of the altruist within the group. However, groups of altruists are more fit

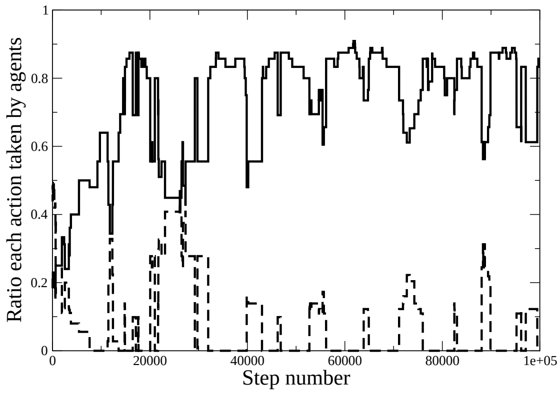

Figure 8. The ratio of each action chosen by agents which received a word expressing a food. The abscissa is step number. The ordinate is the ratio of each action taken by agents. The solid line shows the ratio of the action "going to the cell where words expressing a food". The dashed line shows ratio of the actions "avoiding the cell where words expressing a food". The simulation condition is the same as Fig. 6.

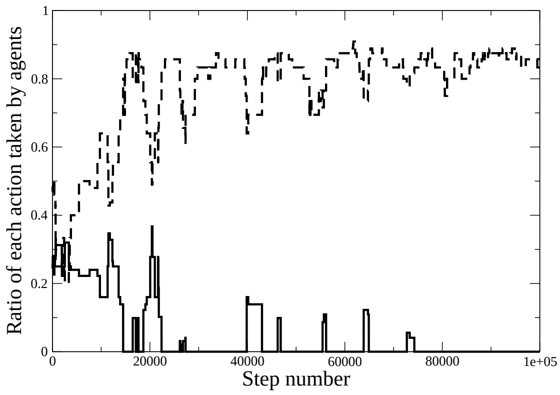

Figure 9. The ratio of each action chosen by agents which received a word expressing a predator. The abscissa and ordinate are the same as Fig. 8. The solid line shows the ratio of the action "going to the cell where words expressing a predator". The dashed line shows ratio of the actions "avoiding the cell where words expressing a predator". The simulation condition is the same as Fig. 6.

than groups of nonaltruists" [4]. If there are only limited number of resources in environment or the carrying capacity of environment is a constant, altruism, such as informing the existence of the resources to others, would decrease the relative fitness of the altruist as Wilson suggested. However, when the amount of resources changes according to its dynamics, some kinds of altruism which increase the carrying capacity by exploiting the dynamics emerges without between group selection as shown in our results showed. Although some studies based on multilevel selection discussed altruism under the constant carrying capability and some studies such as kin selection discussed altruism without considering carrying capability, our result indicate that the capability depends on the type of cooperation and the effect of altruism on the capacity would be one of important keys to understand altruism. In other word, we should pay more attention to the effect of cooperation on the dynamics of agents and environment to discuss altruism.

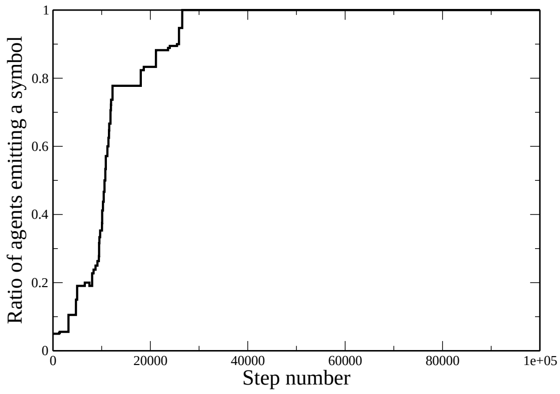

Figure 10. The example of transition of words expressing a food. The abscissa and ordinate are the same as Fig. 6. The solid line shows the number of agents which did not emit words for a food. In this simulation reward of predator is -1 points and death threshold is -900 points.

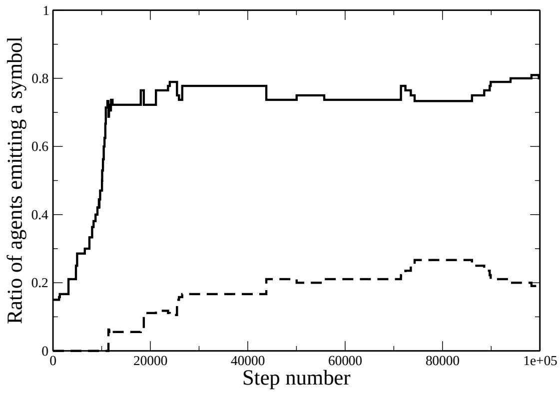

Figure 11. A typical pattern of transition of words expressing a predator. The abscissa and ordinate are the same as Fig. 6. The solid line shows the number of agents which did not emit words for a predator. The dashed line shows the number of agents which emitted '1101' to express a predator. The simulation condition is the same as Fig. 10.

ACKNOWLEDGMENTS

This work has been partially supported by a Grant-in-Aid for Scientific Research on Priority Areas "Emergence of Adaptive Motor Function through Interaction between Body, Brain and Environment" from the Japanese Ministry of Education, Culture, Sports, Science and Technology.

REFERENCES

[1] R. Dawkins, *The selfish gene*, Oxford University Press, (1989).
[2] W. D. Hamilton, 'The genetical evolution of social behaviour. I', *Journal of Theoretical Biology, Vol. 7*, 1–16, (1964).
[3] W. D. Hamilton, 'The genetical evolution of social behaviour. II', *Journal of Theoretical Biology, Vol. 7*, 17–52, (1964).
[4] D. S. Wilson, 'Altruism and organism: Disentangling the themes of multilevel selection theory', *The American Naturalist, Vol. 150*, S122–S134, (1997).
[5] K. R. Foster, T. Wenseleers, and F. L.W. Ratieks, 'Kin selection is the key to altruism', *TREND in Ecology and Evolituin, Vol. 21, No. 2*, 57–60, (2006).
[6] D. S. Wilson and E. O. Wilson, 'Survival of the selfless', *New Scientist*, 42–46, (2007).

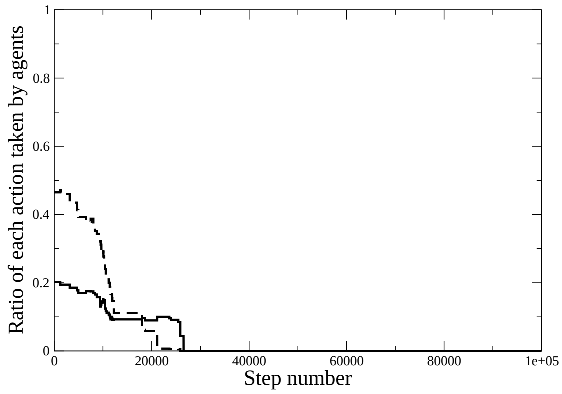

Figure 12. The ratio of each action chosen by agents which received a word expressing a food. The abscissa, ordinate and these lines are the same as Fig. 8. The simulation condition is the same as Fig. 10.

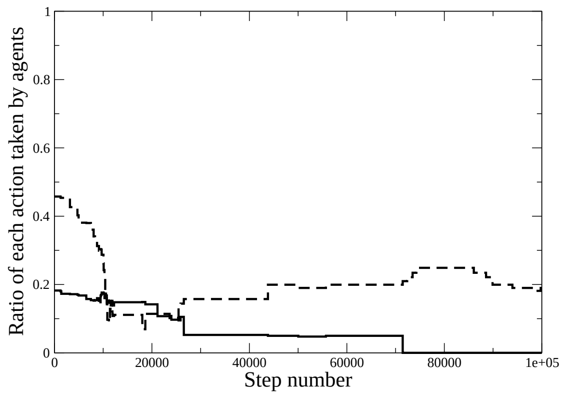

Figure 13. The ratio of each action chosen by agents which received a word expressing a predator. The abscissa, ordinate and these lines are the same as Fig. 9. The simulation condition is the same as Fig. 10.

[7] K. Nakano, Y. Sakaguchi, R. Isotani, and T. Ohmori, 'Self organizing system obtaining communication ability-primitive model for language generation', *Biological Cybernetics, Vol. 58*, 417–425, (1988).
[8] K. Kosmidis, J. M. Halley, and P. Argyrakis, 'Language evolution and population dynamics in a system of two interacting species', *Physica A: Statistical Mechanics and its Applications*, **353**, 595–612, (2005).
[9] Y. Hashizume and J. Nishii, 'Acquisition of common symbols with development of cooperative behaviors', *Proc. of The Tenth International Symposium on Artificial Life and Robotics, CD-ROM*, (2005).

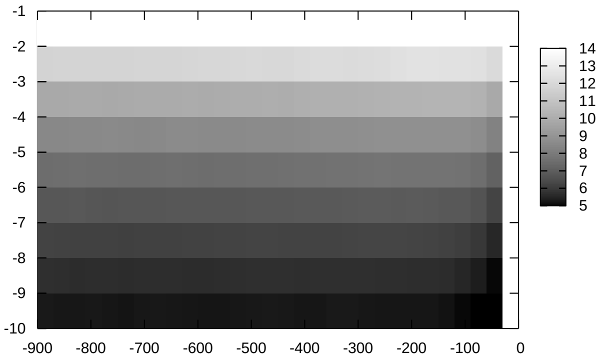

Figure 14. The average of number of agents. The abscissa and ordinate are the same as Fig. 2.

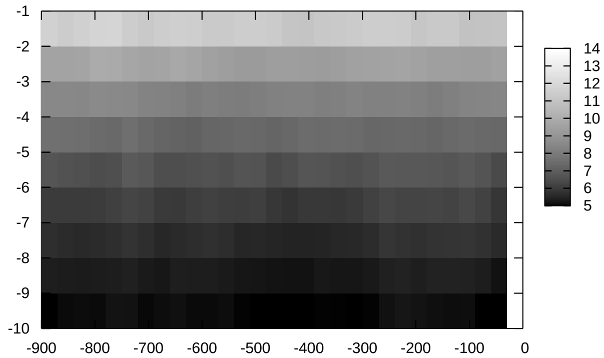

Figure 17. The average of number of agents on condition that agents do not have communication ability. The abscissa and ordinate are the same as Fig. 2.

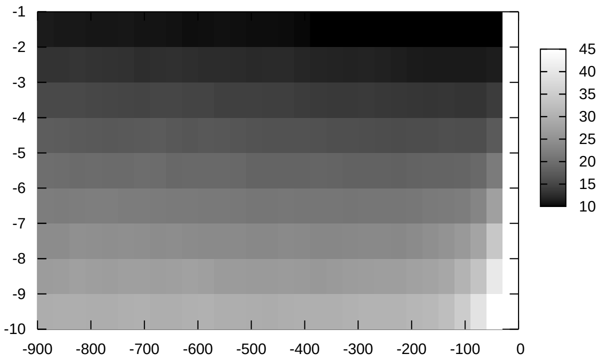

Figure 15. The average of number of foods. The abscissa and ordinate are the same as Fig. 2.

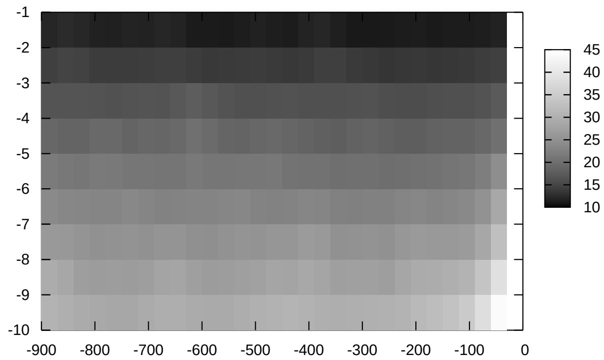

Figure 18. The average of number of foods on condition that agents do not have communication ability. The abscissa and ordinate are the same as Fig. 2.

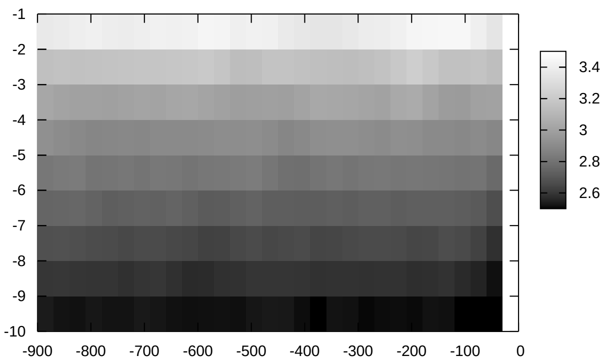

Figure 16. The average of number of predators. The abscissa and ordinate are the same as Fig. 2.

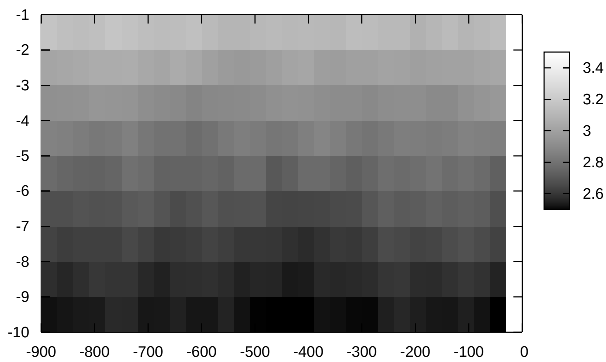

Figure 19. The average of number of predators on condition that agents do not have communication ability. The abscissa and ordinate are the same as Fig. 2.

THE IMPLEMENTATION OF OPTIMIZATION ALGORITHM FOR ENERGY EFFICIENT DYNAMIC AD HOC WIRELESS SENSOR NETWORKS

Mohaned Al. Obaidy [1] and **Aladdin Ayesh** [2]

Abstract. In this work we are presenting the implementation part of our research which explores two of the main Evolutionary Computation techniques which are; Genetic Algorithms (GAs) and Particle Swarm Optimization (PSO) to optimize the energy dissipation in a dynamic Wireless Sensor Network (WSN). We are evolving a hybrid algorithm by applying the GAs in the first phase to divide the sensor network into K-clusters (K-unknown). The output of the first phase will be used as an initial population for the particles in the Swarm which represents the dynamic Sensor Network. GAs proved to be used effectively in the optimization of static Sensor Networks, but for dynamic networks, PSO algorithms are more suitable since the swarms are moving objects by nature. Hence, in this work PSO algorithms are proposed to keep the optimum distances between the sensor nodes during the sensors movement.

1 INTRODUCTION

The integration of sensing, signal processing, and data communication functions allows a WSN to create a powerful platform for processing data collected from the environment. The algorithms and protocols for this kind of network must be able to enable network operation during its initialization and during both normal and exception situations. While the traffic bandwidth requirement is not the main WSN networking issue, the reliability is strongly expected to be fulfilled [5]. Any WSN is deeply involved in and related to the monitored environment, and any change occurring to the surroundings will significantly influence its performance; nevertheless, the network must be able to tolerate and 'survive' any change by implementing proper reactions and adaptation mechanisms sustaining communications for both sensed data and commands [2]. In this work we propose to design an algorithm for a large scale mobile sensors network. This algorithm should provide a robust and energy-efficient communication mechanism which enables the swarms of sensors to move while keeping optimum distances between the sensor nodes.

The rest of this paper will be structured into the follwing sections; Section 2 describes a background ideas and motivation for our work. In section 3, we are explaining the first phase of our proposed algorithm by using GAs to cluster the Sensors Network into independent groups. Section 4 shows the second phase of the proposed algorithm where we use the PSO technique to enable the clusters which are produced in phase-1 to move as Swarms while keeping the optimum distances. In section 5 the implementation of the proposed algorithm is explained by showing some snapshots of the simulation program. Section 6 shows the results discussion as well as comments for the output graphs are presented here including a critical review. Finally in section 7 we concluded our work and its objectives with possible future development and enhancments.

[1] Gulf College, OMAN, email: mohaned@gulfcollegeoman.com
[2] De Montfort University, UK, email: Ayesh@dmu.ac.uk

2 BACKGROUND AND MOTIVATION

Wireless Sensor Networks (WSNs) have gained tremendous importance in recent years because of its potential use in a wide variety of applications. This, along with the unique characteristics of these networks, has spurred a significant amount of research for coming with network protocols specifically tailored for sensor networks [5]. Wireless sensor networks are developing quickly and have been widely used in both military and civilian applications such as target tracking, surveillance, and security management. Since a sensor is a small, lightweight, un-tethered, battery-powered device, it has limited energy [3]. Therefore, energy consumption is a critical issue in sensor networks. We are interested in sensor networks in which a large number of sensors are deployed to achieve a given goal. All data obtained by member sensors must be transmitted to a sink or data collector. The longer the communication distance, the more energy will be consumed during transmission [13]. Direct transmission networks are very straightforward to design but can be very power-consuming due to the long distances from sensors to the sink. Alternative designs that shorten or minimize the communication distances can extend network lifetimes. The use of clusters for transmitting data to a base station leverages the advantages of small transmit distances for most nodes, requiring only a few nodes to transmit far distances to the base station. Clustering means to partition the network into a number of independent clusters, each of which has a cluster-head that collects data from all nodes within its cluster [14]. These cluster-heads then compress the data and send it directly to the sink. The output of GA clustering will be assumed to be the initial population for the Swarms which will represent the dynamic WSN at the later stage of the proposed algorithm. Deployment of mobile swarms can enhance the sensor network in many ways. Firstly, the swarm nodes have much higher hardware capabilities than the sensor nodes. They can provide detailed information of the intended area (e.g. the hot spot). Secondly, the wireless radios of the swarm nodes usually have much longer range and higher channel bandwidth, which can support high quality and delay sensitive multimedia streams. Thirdly, the swarms are mobile [4]. They can be easily directed to the hot spots. A limited number of mobile swarms can easily cover a large scale sensor network. The sensor network can be deployed to cover a very large field due to the low cost of sensor nodes.

3 PHASE-1: GA BASED CLUSTERS INITIALIZATION

Genetic Algorithms are used in phase-1 to generate the optimum clusters distribution for the sensor nodes before moving. In this stage the cluster -heads and its relative members are identified. Algorithm 1 illustrates the basic process in GAs.

Initialization: Generate random population of n chromosomes
while *the stop condition is not satisfied* **do**
  Evaluate the fitness $g(x)$ of each chromosome x in the population;
  while *the new population is not complete* **do**
    Selection: Select two parent chromosomes from a population according to their fitness;
    Crossover: With a crossover probability, crossover the parents to form a new offspring (children);
    Mutation: With a mutation probability mutate new offspring;
    Accepting: Place new offspring in a new population;
  end
  Replace: Use new generated population for further runs;
end
Return: the best solution of the current population;

Algorithm 1: Basic Process in Genetic Algorithms

Once cluster-heads are selected, each regular node connects to its nearest cluster-head. Each node in a network is either a cluster-head or a ”member” of a cluster-head. Each regular node can only belong to one cluster-head. Each cluster-head collects data from all sensors within its cluster and each head directly sends the collected data to the sink. Figure 1 shows an example of clustering. Crossover and Mutation provide exploration, compared with the exploitation provided by selection. The effectiveness of GA depends on the trade-off between exploitation and exploration. [6]

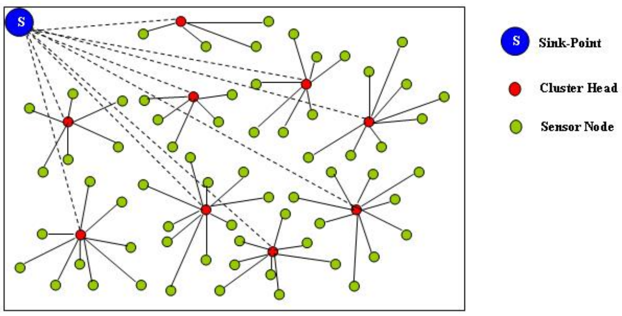

Figure 1. Clustered Sensors Network

The Crossover and Mutation process used in our system are described below:

Crossover: The crossover operation takes place between two consecutive individuals with probability specified by crossover rate. These two individuals exchange portions that are separated by the crossover point. We use in this paper one-point crossover. The following is an example of crossover:

Indv1: 1 1 1 0 0 1 0 1
Indv2: 1 0 1 1 1 1 1 0
$\Uparrow$
Crossover point

After crossover, two offspring are created as shown below:

Child1: 1 1 1 0 1 1 1 0
Child2: 1 0 1 1 0 1 0 1

If a regular node becomes a cluster-head after crossover, all other regular nodes should check if they are nearer to this new cluster-head. If so, they switch their membership to this new head. This new head is detached from its previous head. If a cluster-head becomes a regular node, all of its members must find new cluster-heads. Every node is either a cluster-head or a member of a cluster-head in the network.

Mutation: The mutation operator is applied to each bit of an individual with a probability of mutation rate. When applied, a bit whose value is 0 is mutated into 1 and vice versa. An example of mutation is as follows:

Indv: 1 1 1 1 1 1 0
$\Downarrow$ $\Downarrow$
Indv: 1 1 1 0 1 1 1

3.1 Chromosome Representation of Distance-Head Problem

In order to find appropriate cluster-heads is critically important to minimizing the distance. We use binary representation in which each bit corresponds to one sensor. A ”1” means that corresponding sensor is a cluster-head; otherwise, it is a regular node. In the following example:

s1	s2	s3	s4	s5	s6	s7	s8
1	0	0	1	0	1	0	0

Individual nodes s1, s4 and s6 are cluster-heads. The remaining nodes are regular sensors. The initial population consists of randomly generated individuals. GA is used to select cluster-heads. Each regular node uses a deterministic method to find its nearest cluster-head.
In our proposed algorithm we modified the basic GA in a way that in case of any cluster-head remain unconnected with any regular sensor then its state must be changed to be a regular sensor and should be linked with the nearest cluster-head available in the field. The proposed algorithm is shown in Algorithm 2.[10]

3.2 Fitness Function: Distance-Number of Heads Rule

The total transmission distance is the main factor we need to minimize. In addition, the number of cluster heads can factor into the function. Given the same distance, fewer cluster heads result in greater energy efficiency as cluster heads drain more power than non-cluster-heads. Thus, each individual is evaluated by the following combined fitness components:

Initialization: Generate random population of n chromosomes
while *the stop condition is not satisfied* **do**
  if *cluster-head not connected to any sensor-node* **then**
    change cluster-head state into regular sensor;
    find the nearest cluster-head to be connected with;
  end
  Evaluate the fitness $g(x)$ of each chromosome x in the population;
  while *the new population is not complete* **do**
    Selection: Select two parent chromosomes from a population according to their fitness;
    Crossover: With a crossover probability, crossover the parents to form a new offspring (children);
    Mutation: With a mutation probability mutate new offspring;
    Accepting: Place new offspring in a new population;
  end
  Replace: Use new generated population for further runs;
end
Return: the best solution of the current population;

Algorithm 2: Modified GA Algorithms

$$Fitness = w * (D - distance_i) + (1 - w) * (N - H_i)$$

where *D* is the total distance of all nodes to the sink, $distance_i$ is the sum of the distances from regular nodes to cluster-heads plus the sum of the distances from all cluster-heads to the sink; H_i is the number of cluster-heads; N is the total number of nodes; and w is a predefined weight. Except for $distance_i$ and H_i, all other parameters are fixed values in a given topology. The shorter the distance, or the lower the number of cluster-heads, the higher the fitness value of an individual is. Our GA tries to maximize the fitness value to find a good solution. The value of w is between 0 and 1, and it is application-dependent. It indicates which factor is more important to be considered: distance or the cost incurred by cluster-heads. At one extreme, if $w = 1$, we optimize the network only based on the communication distance. If $w = 0$, only the number of cluster heads is considered.

4 PHASE-2: PSO BASED MOVABLE CLUSTERS

Particle Swarm Optimization was first proposed by [7]. In this method a set of potential solutions are called particles that are initialised randomly. Each particle will have a fitness value, which will be evaluated by the fitness function to be optimised in each generation. Each particle knows its best position $pbest$ and the best position so far among the entire group of particles $gbest$. The particle will have velocities, which direct the flying of the particle. In each generation the velocity and the position of the particle will be updated. The velocity and the position update equations are given below as (1) and (2) respectively.

$$v_i^{k+1} = wv_i^k + c_1 rand_1 * (pbest_i - s_i^k) + c_2 rand_2 * (gbest - s_i^k) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$

The parameters used in equations 1 and 2 are described in Table 1.

In recent times, there has been a number of improvements to the original PSO [8]. We have explored different versions of PSO where the extension to the original algorithm is distinct from each other. Following PSO versions are studied in this paper:

[htbp]

Table 1. The parameters for PSO velocity and position update

Parameter	Description
v_i^k	velocity of particle i at iteration k
w	inertia weight
v_i^{k+1}	velocity of particle i at iteration $k+1$
c_j	acceleration coefficients j=1,2
$rand_i$	random number between 0 and 1 i=1,2
s_i^k	current position of particle i at iteration k
$pbest_i$	$pbest$ of particle i
$gbest$	$gbest$ of the group
x_i^{k+1}	position of the particle i at iteration $k+1$

4.1 PSO - Time Varying Inertia Weight (TVIW)

PSO-TVIW model is the same basic PSO algorithm with inertia weight parameter is varying with time from 0.9 to 0.4 and the acceleration coefficient is set to 2. This model is proposed by [12]. The time varying inertia weight is mathematically represented as follows:

$$w = (weight - 0.4) * \frac{(MAXITER - iter)}{MAXITER} + 0.4 \quad (3)$$

Where, MAXITER is the maximum iteration allowed, $iter$ is the current iteration number and $weight$ is a constant set to 0.9.

4.2 PSO-Time Varying Acceleration Coefficients (TVAC)

PSO-TVAC model proposed by [1]. In this model, the time varying acceleration coefficients (TVAC) are used in such a way that; c_1 varies from 2.5 to 0.5 and the c_2 varies from 0.5 to 2.5. Here the cognitive component is reduced and social component is increased by changing c_1 and c_2. The large cognitive component and the small social component in the initial stages of the algorithm helps the particle to wander around the search space. However, the small cognitive component and large social component at the later stages of the algorithm helps the particle to converge to the global optima. TVAC is mathematically represented as follows:

$$C_1 = (C_1min - C_1max)\frac{iter}{MAXITER} + C_1min \quad (4)$$

$$C_2 = (C_2min - C_2max)\frac{iter}{MAXITER} + C_2min \quad (5)$$

In Equations 4 and 5 c_1min and c_2min are constants set to 0.5, c_1max and c_2max are also constants set to 2.5. Thus, in this algorithm as the $iter$ progresses, c_1 varies from 2.5 to 0.5 and c_2 varies from 0.5 to 2.5.

4.3 Hierarchical Particle Swarm Optimizer with Time Varying Acceleration Coefficients (HPSO-TVAC)

In this method the particle behaviour will not be influenced by the previous velocity term of Equation 1. Due to non-influence of previous velocity, re-initialisation of velocity is used when the velocity stagnates in the search space [1]. Therefore, a series of particle swarm optimisers are automatically generated inside the main particle swarm optimiser according to the behaviour of the particle in the search space, until the convergence criteria is met.

The reinitialisation of velocity is set proportional to $Vmax$. The pseudocode for reinitialising velocity is as follows:

$$
\begin{aligned}
& v_i^{k+1} = c_1 rand_1 * (pbest_i - s_i^k) + c_2 rand_2 * (gbest - s_i^k) \\
& \qquad if(v_i^{k+1} == 0) \\
& \qquad\qquad if(rand_1() < 0.5) \\
& \qquad\qquad v_i^{k+1} = rand_2() * v \\
& \qquad\qquad else \\
& \qquad\qquad v_i^{k+1} = -rand_3() * v \\
& \qquad\qquad endif \\
& \qquad endif \\
& v_i^{k+1} = sign(v_i^{k+1}) * min(fabs(v_i^{k+1}, v_{max}))
\end{aligned}
$$

where $rand_i()$, i= 1, 2, 3 are separately generated uniformly distributed random numbers in the range [0,1] and v is the reinitialisation velocity. The effect of **HPSO** along with **TVAC** (hence, **HPSO-TVAC**) on clustering of sensor networks can be observed through simulations.

4.4 Particle Swarm Optimisation with Supervisor-Student Model (PSO-SSM)

In this method [16] proposed PSO-SSM to achieve low computational costs. The algorithm introduces a new parameter called momentum factor (mc) to update the positions of particles. In this algorithm, they also proposed a different velocity updation mechanism from the conventional PSO algorithms. Here velocity is updated only if each particle's fitness at the current iteration is not better than that of previous iteration. The velocity serves as a navigator (supervisor) by getting the right direction, while the position (student) gets a right step size along the direction. The velocity and the position are modified using the following equations:

$$v_i^{k+1} = v_i^k + c_1 rand_1 * (pbest_i - s_i^k) + c_2 rand_2 * (gbest - s_i^k) \quad (6)$$

$$x_i^{k+1} = (1 - mc) * x_i^k + mc * v_i^{k+1} \quad (7)$$

5 IMPLEMENTATION AND EXPERIMENTATION

5.1 Energy Model for Optimisation

We are studying the impact of the transmission range of sensor nodes and positioning of the sink in minimising the communication energy in a sensor network. The important components of each sensor are the data and control processing unit and the radio for communication. The microprocessor used in the processing unit should be energy efficient with less energy consumption. The energy dissipation in the radio depends on the different characteristics of the radio. The energy model used in this work is adopted from [14, 11, 15] and summarised here. The energy dissipation for transmitting b bits to d distance is shown in Equation 8.

$$E_{tx}(b, d) = E_{elec} \times b + E_{amp} \times b \times d^2 \quad (8)$$

The energy dissipation in a node to receive b bits of data is shown in Equation 9.

$$E_{rx}(b) = E_{elec} \times b \quad (9)$$

Where E_{elec} is the radio energy dissipation and E_{amp} is the transmition amplifier energy energy disipation. Energy consumption of a wireless sensor node transmitting and receiving data from another node at a distance d can be divided into two main components: Energy used to transmit, receive and amplify data and energy used for processing the data, mainly by the microcontroller. Leakage current can be as large as a few mA for the microcontroller, and the effect of leakage current can be neglected for higher frequencies and lower supply voltage. Assuming the leakage current as negligible, the total energy loss for the sensor system due to the distance E_{dd} can be calculated according to Figure 2 using the following equation:

$$E_{dd} = \left(\sum_{j=1}^{k} \sum_{i=1}^{n_j} (d_{ij}^2 + \frac{D_j^2}{n_j}) \right) \quad (10)$$

For more details about the derivation and proof refer to [11].

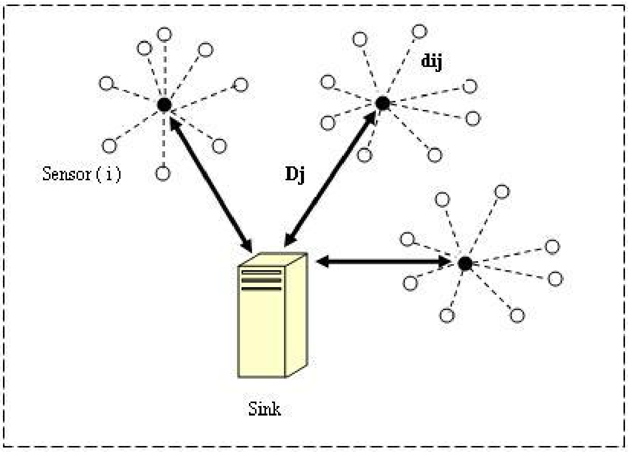

Figure 2. Energy Model for distance based Sensor Network

5.2 Experiments and Simulation

In this section, we explore the use of GAs and PSO to solve the distance minimization problem for dynamic sensor networks.

Phase-1:
To implement our algorithm, we used Java-Applet as a programming environment to simulate an experiment with 100 generated random nodes in a simulated 2-D environment with two different sink positions located at (0,0) and (100,100). As a tuning parameters for GAs, we used the parameters given in Table 2.

Table 2. The GA parameters settings

Parameter	Value
Population size	100
Selection type	Proportional
Crossover rate	0.7
Crossover type	one point
Mutation rate	0.005
Generation size	1000

We explored three case studies. They are:

- **case 1:** when the sink point is located at (0,0) (i.e. the upper left corner) and w is set 1.0, Figure 3. This network distribution is suitable when the application environment is inhospitable, which will be not safe to allocate the sink-point (i.e. data collector) within the field area like some military applications or earthquake observations, etc.

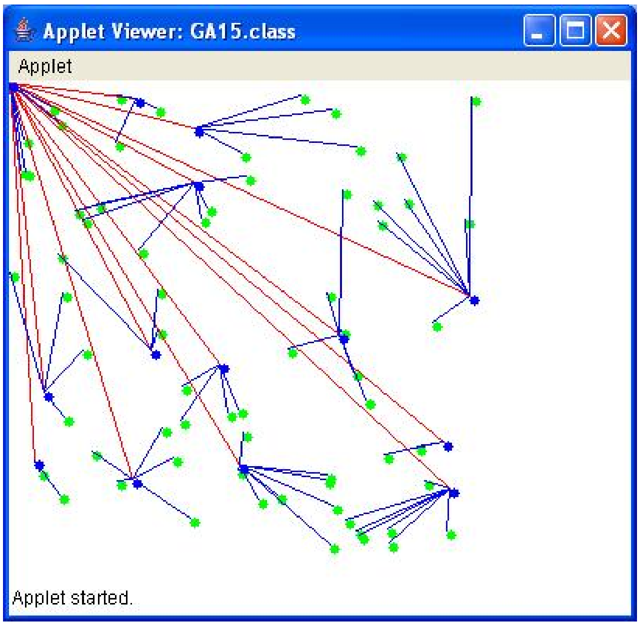

Figure 3. Clustered network when sink point at (0,0)

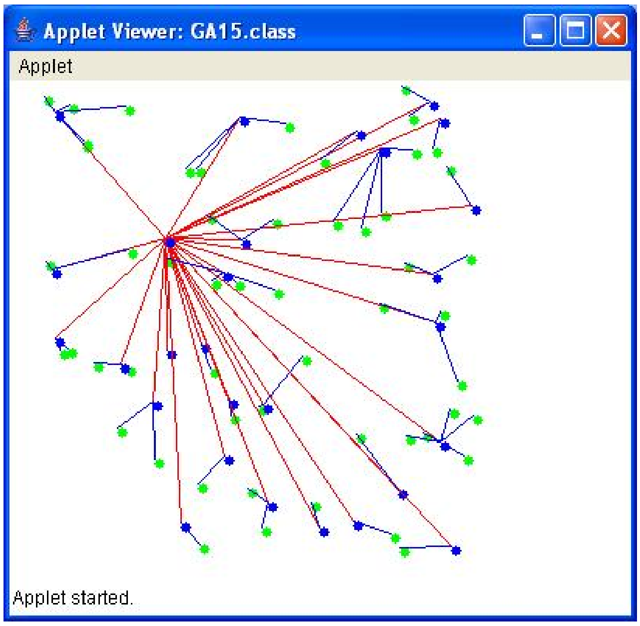

Figure 4. Clustered network when sink point at (100,100)

- **case 2:** when the sink point is located at (100,100) and w set to 0.8, Figure 4. This network distribution is more suitable when the sensor nodes are distributed around a centralized safe area where the sink-point can receive the data in a wider circular range and from different directions. For example the Mobile networks.
- **case 3:** when w is set to 0, Figure 5. This figure describes the situation when the number of cluster-heads is only considered in the fitness function. Although this is not realistic in our problem, but it verifies the effectiveness of our algorithm because, as expected, the optimal number of heads is 1.

Phase-2:
Refering to Equation (10), we can conclude that by reducing the distance from a node to the cluster-head and the cluster-head to the sink we can minimise the energy dissipation in a sensor network. In our simulation, we cluster the nodes taking into consideration that each node can transmit or receive data from all the other nodes. Thus, nodes considered in this network do not have transmission range constraint. Sensors are clustered using entirely distance based Equation (10). The fitness function for this method is as follows [9]:

$$Fitness = min\left(\sum_{j=1}^{k}\sum_{i=1}^{n_j}(d_{ij}^2 + \frac{D_j^2}{n_j})\right) \tag{11}$$

where,

$$\textstyle\sum_{j=1}^{k}(n_j + k) = N.$$

N is the number of nodes in a network. For our simulations,

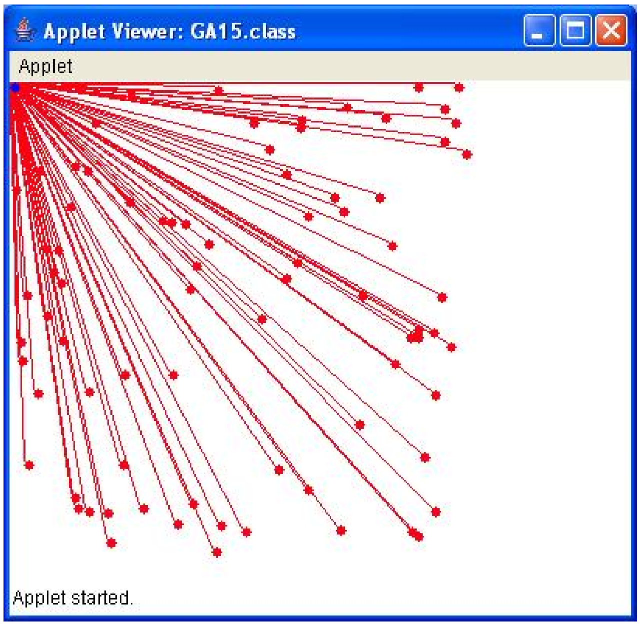

Figure 5. Clustered network when w=0

we used 100-node networks that are uniformly distributed in a 2-Dimensional problem space [0:100,0:100]. We have studied the impact of sink location on the fitness value of the PSO algorithms. In one set of simulations we considered the sink-point to be located at the center of the network (50,50). In another set of simulations we considered the sink-point to be located remotely at (50,180). For both simulations we use the same set of nodes. The maximum number of generations we were running was 1000. The parameters used in the simulations are tabulated in Table 3.

Table 3. Initialisation and Parameters Range

Parameter	Range
Population size	100
$MAXITER$	1000
v_{max}	100
x_{max}	100
v range	0-100
x range	0-100

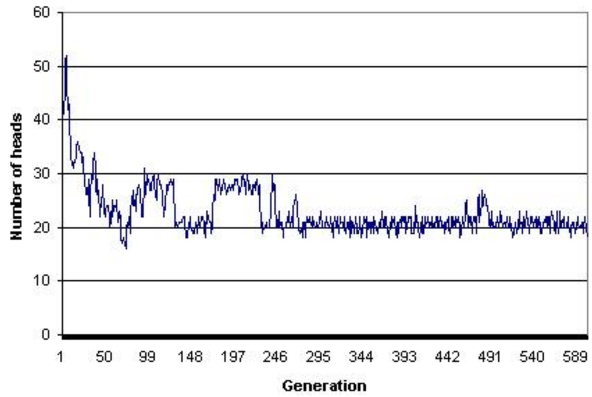

Figure 7. Number of Cluster heads over generations using GAs

6 CRITICAL REVIEW AND RESULTS

Our proposed approach was able to find quickly the optimal solutions. For a 100-node problem, a good solution can be achieved after around 130 generations, as shown in Figure 6 which is relatively a small number of generations in such applicatons. The fitness value

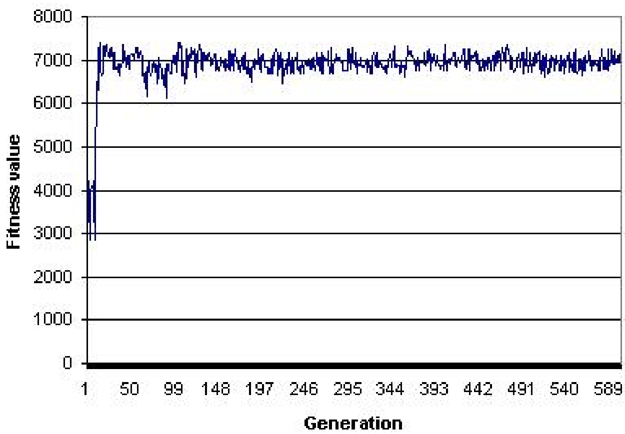

Figure 6. Fitness values over generations using GAs

is greatly enhanced after 100 generations due to the selection of the best fitness chromosomes to be used in the next generation. In Figure 7, number of cluster heads decreases over generations to reach around 25% from the overall number of nodes in the network. This verifies the effectiveness of our algorithm because, as expected, the total distance will be minimized as the number of heads decreases.

Experiments indicate that the scaling window plays an important role in the quality of the solution found. When a single node is near to the sink, that node itself becomes a cluster-head and sends data directly to the sink. Experiments also show that nodes near to the sink are more likely become cluster-heads than those far away. More cluster-heads are needed when a sink is close to the center of a network than when it is located at a network corner. This observation is expected because when the sink is at the center, all regular nodes are located around the sink. As a result, cluster-heads tend to be distributed around the sink.

In this work we observed the performance in terms of quality of the average optimum value for 10 trials to the **PSO-SSM** and **PSO-TVIW** models which are described earlier. We chose these two methods for the following reasones; the **PSO-SSM** model is the only model which has the ability to stop particles from moving beyond the boundary of the problem space, that is under the influence of mc parameter in it. The **PSO-TVIW** model is almost similar to the basic PSO algorithm with just the inertia weight varying with time from 0.9 to 0.4. From the graph shown in Figure 8 we can conclude that **PS-TVIW** convergence is slower as compared to the **PSO-SSM** algorithm. This was due to constant acceleration co-efficients used in this model which affects the rate of convergence.

Simulation results show that the proposed approach is an efficient and effective method for solving this problem with respect to distance minimization.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we propose the use of GAs to minimize the communication distance in a sensor network by dividing it into clusters and the use of PSO to make this network moves as a Swarm keeping the optimum distances between the sensors while they are moving. Our proposed approach starts by taking random selecting nodes in a network to be used as a cluster-heads. The algorithm then starts to find an appropriate number of cluster-heads and their locations by adjusting cluster-heads based on fitness function. We also explored the results of the performance evaluation of four extensions to the standard Particle Swarm Optimization algorithm in order to reduce the energy consumption in Wireless Sensor Networks. Communication distance is an important factor to be reduced in sensor networks. We have simulated two models; the Supervisor-Student Model (PSO-SSM) and the time varying Inertia Weight (PSO-TVIW) model. In the (PSO-SSM) model the new parameter introduced called the mo-

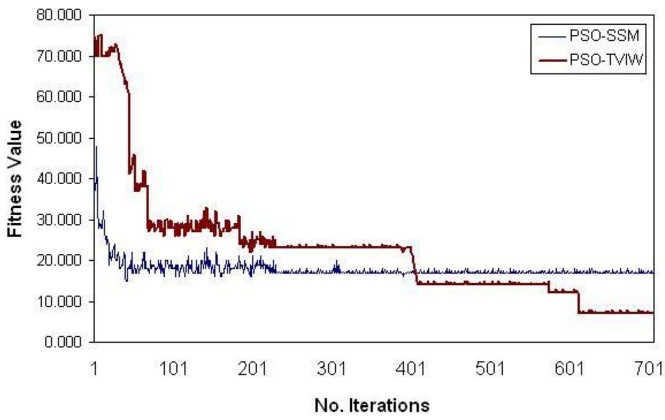

Figure 8. Convergence for the PSO-SSM and PSO-TVIW Models

mentum factor mc to update the position of particles. Also here the velocity is updated only if each particle's fitness at the current iteration is not better than that of previous iteration. Hence the computational costs for this algorithm will be decreased. An important modification proposed is to use boundary checking for re-initialization of particle which moves outside the set boundary. We can also conclude that (PSO-TVIW) convergence is slower as compared to other algorithm. As a future work, our program can be upgraded to cover the two other models described in this paper, then a comprehensive comparison could be done to analyze the behavior of the particles within each case.
We plan to extend the problem on hand by considering a hierarchical structure where a cluster-head can have a super cluster-head which sends data directly to the sink.

REFERENCES

[1] S. Halgamuge A. Ratnaweera and H. Watson, 'Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients', *Evolutionary Computation, IEEE Transactions*, **8, No.3**, 240–255, (2004).

[2] S. Bandyopadhyay and E. J. Coyle, 'An energy efficient hierarchical clustering algorithm for wireless sensor networks', *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, (2003).

[3] Lucille Verbaere Armin Wellig C. Laurent, Didier Helal and Julien Zory, 'Wireless sensor networks devices: Overview, issues, state of the art and promising technologies', *ST Journal of Research*, **4, No.1**, (June 8, 2007).

[4] Mario Gerla and Kaixin Xu, 'Multimedia streaming in larg-scale sensor networks with mobile swarms', *SIGMOD Record*, **32, No. 4**, 72–76, (December, 2003).

[5] Y. Sankarasubramaniam I. Akyildiz, W. Su and E. Cayirci, 'A survey on sensor networks', *IEEE Communications Magazine*, 102–114, (August 2002).

[6] Shiyuan Jin, Ming Zhou, and Annie S. Wu, 'Sensor network optimization using a genetic algorith', *Proceedings of the 7th World Multiconference on Systemics, Cybernetics, and Informatics*, (July 2003).

[7] J. Kennedy and E. R.C., 'Particle swarm optimization', *IEEE Inteniational conference on Neural Networks, Perth, Australia*, 1942–1948, (1995).

[8] M. Obaidy and A. Ayesh, 'Energy efficient pso-based algorithm for optimizing autonomous wireless sensor network', in *European Simulation and Modelling (ESM'2008) Conference. EUROSIS, Le Havre, France*, (2008).

[9] M. Obaidy and A. Ayesh, 'Optimizing autonomous mobile sensors network using pso algorithms', in *Proceedings of the International Conference on Computer Engineering & Systems (ICCES'08), Egypt*, (2008).

[10] M. Obaidy, A. Ayesh, and A. Sheta, 'Optimizing the communication distance of an ad hoc mobile sensor networks by genetic algorithms', in *Proceedings of the Forth International Workshop on Advanced Computation for Engineering Applications (ACEA08), Jordan*, pp. 17–23, (July 2008).

[11] S. Halgamuge S. M. Guru, A. Hsu and S. Fernando, 'An extended growing self-organising map for selection of clustering in sensor networks', *International Journal of Distributed Sensor Networks*, **1, No.2**, (2005).

[12] Y. Shi and R. Eberhart, 'Empirical study of particle swarm optimization', *Proceedings of the Congress on Evolutionary Computation, CEC 99*, **3**, 1950, (1999).

[13] A. Chandrakasan W. R. Heinzelman and H. Balakrishnan, 'Energy efficient communication protocol for wireless micro-sensor networks', *In Proceedings of the Hawaii International Conference on System Science, Maui, Hawaii*, 3005–3014, (2000).

[14] A.P. Chandrakasan W. R. Heinzelman, 'An application-specific protocol architecture for wireless micro-sensor network', *IEEE Transactions on Wireless Communications*, **1,No.4**, 660–670, (2002).

[15] A. Wang and A. Chandrakasan, 'Energy-efficient dsps for wireless sensor networks', *Signal Processing Magazine, IEEE*, **19, No.4**, 68–78, (2002).

[16] Z. Qin Y. Liu and X. He, 'Supervisor-student model in particle swarm optimization', *Evolutionary Computation, CEC2004 Congress*, **1**, 542–547, (2004).

Crew Intelligence Systems for Digital Objects Preservation

Josep Lluis de la Rosa[1], Esteve del Acebo[1], Albert Trias[1], Silvana Aciar[1], and Hugo Quisbert[2]

Abstract - Crew Intelligence Systems is a family of very simple algorithms for different classes of complex optimization problems in static and dynamic environments by means of reactive multi agent systems. Crew Intelligence systems are loosely inspired from the behavior shown by a staff of bartenders when serving drinks to customers in a bar or pub. In this paper we improve them by letting the bartenders also call (shout) for help, and we adapt them to Digital Objects Preservation, where agents explore file systems looking for victims, digitals objects that need change of format, or any other transformation to be preserved. When they find someone they "shout" so that agent mates can hear it. The louder the shout, the most important or urgent the finding. Louder shouts can also refer to closeness. We perform several experiments to show that this system works very scalably, showing that heterogeneous teams of agents outperform homogeneous ones over a wide range of task complexity. Finally, a properly designed combination of heterogeneous agents is more scalable when confronted with uncertain maps of digital objects to be preserved

1. INTRODUCTION

The term Swarm Intelligence, which has garnered much attention, arose in the late 1990s in the Artificial Intelligence, Robotics, Artificial Life, and Distributed Problem Solving communities, inspired by the observation of social insect colonies. A commonly accepted definition of it is: "the property of a system whereby the collective behaviors of (unsophisticated) agents interacting locally with their environment cause coherent functional global patterns to emerge". The chief paradigm of Swarm Intelligence is an ant colony. In it, individual ant behavior is controlled by a small set of very simple rules, but their interactions (also very simple) with the environment allow them to solve complex problems such as finding the shortest path from one point to another. Ant colonies (and we could say the same about human beings) are intelligent systems with great problem-solving capabilities, formed from a quantity of relatively independent and very simple subsystems that do not show individual intelligence. This is the "many dummies make a genius" phenomenon of emergent intelligence.

Swarm Intelligence problem-solving techniques present several advantages over more traditional ones. On one hand, they are cheap, simple and robust; on the other hand, they provide a basis for exploring collective (or distributed) problem-solving without centralized control or the provision of a global model. Over the last few years, they have been used in the resolution of a very heterogeneous class of problems. Two of the most successful Swarm Intelligence techniques currently in use are Ant Colony Optimization [1] and Particle Swarm Optimization [2]. Ant Colony Optimization techniques, also known as Ant Systems, are based on ants' foraging behavior, and have been applied to problems ranging from the determination of minimal paths in TSP-like problems to network traffic rerouting in busy telecommunications systems. Particle Swarm Optimization techniques, inspired from the way a flock of birds or a school of fish moves, are general global minimization techniques that deal with problems in which a best solution can be represented as a point or surface in an n-dimensional space. Other Swarm Intelligence applications include collective robotics, agent navigation, planetary mapping, streamlining of assembly lines in factories, coordinated agentic transport, and banking data analysis. For more details on self-organization and Swarm Intelligence theory and applications, please refer to [3], [4], [5], [6], [7] and [8].

No doubt, Swarm Intelligence techniques have proved its usefulness over the last years. Nevertheless, in our opinion, their applicability and effectiveness is somewhat limited by the simplicity of the individual agents in the swarm. In the typical Ant Colony Optimization systems, for example, ants behavior is purely reactive and communication between ants is only allowed through the environment, in the form of a pheromone trail. One can't help but wonder whether it would be possible to increase the individual communication and problem solving capabilities of the agents in a Swarm Intelligence system, while at the same time maintaining the desirable features of cheapness, locality, decentralization, simplicity and robustness and what impact would it have in the overall behavior of the system.

It turns out to be that it is possible to find such systems in the real world, especially in those situations where people have to coordinate themselves in a highly dynamic environment in order to solve some kind of scheduling process. Examples are a vessel crew, a staff of bartenders serving pints in a pub or a soccer team. These kind of systems are characterized by highly dynamic environments where tasks of different classes quickly appear and disappear and have to be carried out in a timely fashion. Coordination between people in this kind of systems is very important, but is not attained, typically, by means of some centralized global procedure. The behavior of the individual agents is mainly reactive (they react to the appearance and disappearance of tasks) but, at the same time, and this differentiate them from classical Swarm Intelligence systems,

[1] EASY Innovation Center University of Girona, Campus de Montilivi, E17071 Girona, Catalonia (EU) +34 972 418478, peplluis@eia.udg.edu, acebo@ima.udg.edu, albert.trias@gmail.com, saciar@eia.udg.edu
[2] Luleå University of Technology, University Area, 97187, Luleå (Sweden) +46 920 491810, Hugo.Quisbert@ltu.se

they make use of their "human" abilities (complex communication, reasoning, planning…) to coordinate themselves and increase the problem solving effectiveness of the system. We have chosen to Christianize those kinds of empowered Swarm Intelligence Systems with the name of Crew Intelligence Systems, though early versions were named as Bar Systems [9]. Crew Intelligence are reactive multi agent systems whose behavior is loosely inspired in that of a crew of bartenders. Three traits distinguish them:

- They are well suited for finding approximate solutions for large and complex scheduling real-time problems in highly dynamic environments.
- The behaviours of individual agents are directed towards the maximization of a local affinity function. This individual behavior results in the whole system tending to the minimization of a global cost function.
- Individual agents are endowed with more or less complex communication and local planning abilities which increase the problem solving capabilities of the system.

S&A is a type of Crew Intelligence System, reactive multi-agent systems whose behavior is loosely inspired on that of a staff of bartenders, and can be enclosed in the broader class of Swarm Intelligence systems. We will explore its application to Digital Objects Preservation. The preservation of digital objects is a new issue that comes up with the continuous growth and advance of information, that is created and stored digitally Few solutions have been given to solve how this information will be accessed in the future, even within the next decade or so [10]. Even if the information itself survives over time, the hardware and the software to access it may not. As a result "rescue information" is required to ensure ongoing access to digital preservation for as long as it is requires and for whatever. In this paper is presented how digital objects (text files, photography, audio, etc) can be "rescued" for Long-Term Digital Preservation using solutions inspired from rescue robots and swarm intelligence like algorithms.

This paper is organized as follows: section 2 will describe the Digital Objects Preservation problem and the type of algorithms it needs; the concept of Shout and Act as a type of Crew Intelligence System will be presented and formalized in section 3; in sections 4 and 5 we will present the results of preservation experiments, with several teams of agents and varying complexity of preservation grids; and in section 6 we present our conclusions.

2. THE PROBLEMS OF PRESERVING DIGITAL OBJECTS

Hardware and software obsolescence is jeopardizing the availability and accessibility to digitally recorded information. Moreover, large amounts of digital information have been created and this growth continues exponentially making digital preservation complex. Digital Preservation aims to alleviate threats caused by digital obsolescence and the rapid growth of information. [10] Digital Preservation can be approached as a technical issue using Agents. The most important demand from digital preservation is [14]] that the number of digital objects to be preserved in growing exponentially, and we need algorithms and solutions that could offer linear complexity, better if they offered 0 "plain" complexity.

There are many aspects that determine the difficulty of Digital Objects preservation: planning is done in real time, and heterogeneous multiagent systems work in dynamically changing and even hostile environments in which new and complex scenarios constantly appear. Softbots (they will be referred equally as "agents") interact with totally alien cognitive agents, for example, human beings, with the need for monitoring and constant optimization of scarce resources.

We conceive three approaches for the application of agents to digital preservation:

- First, agents work like recommenders. For example agents can work together for more accurate appraisal of the digital objects to be saved. This approach will help users share their points of view and solutions in a Web 3.0 way, as the automation of web 2.0 approaches to digital preservation, providing with easier tools for preservation. An example of this type of agents is Giulia, shown in [14]].
- Second, agents work like rescue robots. Agents look for digital objects, the victims, with expected short life (for example, format problems) and try to save them. This is expected to contribute with scalability to the exponentially growing complexity of digital preservation.
- Third, agents are the digital objects to be preserved. Agents look for surviving and compete for being preserved. The log of formats is kept inside the agent,. who negotiates with web services or other agents for being itself preserved, and earn credits. This is a native application of agents.

In this paper we will develop the second approach.

Well show how Crew Intelligence Systems are ideal to deal with Digital Preservation requirements, as long as agents are specifically designed to support the communication skills that these systems require.

3. DESCRIPTION OF THE ALGORITHM

For the sake of specificity we are going to call Shout and Act (S&A) the specific Crew Intelligence System algorithm Agents are generally exploring the file system, and will shout when they find an item of interest. Should the shouting agent modulate its volume to indicate distance, or do you simply mean that closer agents are "heard" more loudly? Both two cases apply. The volume of the transmitted shout corresponds to the importance of the finding; the volume received also depends on the distance. The more agents in a position, the more likely it will be for a victim to be at that position. Agents will be of several types, each with heterogeneous skills and capabilities, shouting to (calling) other agents when they detect a potential victim. Those agents with more sensing capabilities tend to be more costly and therefore less numerous. They might follow the hints given by the shouts of inferior types of agents, and shout to summon even more superior types of agents when they themselves detect something.

Let us now describe S&A from the Crew Intelligence Systems point of view [9]: Anyone who has tried to get served a pint in a bar crowded with customers will have had more than enough time to wonder with boredom about the method used by waiters, if there is one, to decide which customer to pay attention to at each time. Sometimes, one may have to wait a long time before being served, even if shouting for the waiter. . Details like the area where the customer is located, his/her sex, whether the waiter knows him/her, and whether the waiter likes the customer's appearance determines to a large extent the way in which orders are served.

Let us examine the situation from the bartenders' point of view: many customers are ordering drinks at once, new ones arrive constantly, and the bartenders have to do all they can to serve them. Of course, they cannot do this in a random way; they have to try to maximize some kind of utility function, which will typically take into account aspects such as average service time, average service cost, and average customer/boss satisfaction. They will have to pay attention, then, to facts such as that some of them can prepare certain drinks more quickly or better than others, that the order in which the drinks are served influences the time or the total cost of serving them, and also that moving from one place to another takes time. All of this must be done without forgetting, on one hand, that the order in which orders take place has to be respected as much as possible and, on the other hand, that they have to try to favor the best customers by giving them preferential attention and keeping them waiting for a shorter time.

The problem is not at all trivial, and we proved it to be NP-hard in [9]. Bartenders have to act in a highly dynamic, asynchronous and time-critical environment, and no obvious greedy strategy (such as serving the best customer first, serving the nearest customer first, or first serving the customer who has arrived first) gives good results. Nevertheless, a staff of good bartenders usually can manage to serve customers in such a way that the vast majority of them are, more or less, satisfied. The way they accomplish the task seems to have little to do with any global planning or explicit coordination mechanisms but, arguably, with trying to maximize, every time they choose a customer to serve, some local utility function that takes into account aspects like the importance of the customer, the cost for the waiter of serving him/her, and the time that he/she has been waiting for service.

3.1 Outline the Behavioral Rules in Spirit of the Algorithm Formal Approach

We must design agents (softbots) with the following properties from soft-agency:

a) *Agents must be autonomous and proactive*, able to perform desired tasks in unstructured environments without continuous human guidance. A high degree of autonomy is particularly desirable because communication delays and interruptions are unavoidable. They must be able to:
    - Gather information about the environment.
    - Move either all or part of themselves throughout its operating environment without human assistance.
    - Avoid situations harmful to digital objects, property, or themselves.
    - Learn or gain new capabilities, including adjusting strategies for accomplishing tasks and adapting to changing environments.

b) *Agents must be self aware or introspective;* this is a soft-agency property that requires an agent to observe itself, optimize its behavior to meet its goals, and communicate its current limitations to others. Some examples of self-awareness in a system are:
    - State diagnosis: credit for the application of web services, available memory, available disk space, etc.
    - Ongoing activities: serving users, appraising of curating objects.
    - Choosing actions under external and internal constraints.
    - Knowledge and lack of knowledge.
    - Purpose, intentions, hopes, fears, likes, dislikes.
    - Mental state, e.g. long term goals, and beliefs (hard agency).

c) *Agents must be social, able to communicate directly and indirectly.* They must be able to negotiate among themselves to surmount unforeseen properties of a master plan.

d) *Agents must be heterogeneous*, preferably with complementary skills, expected to work better together than separately, though working as a team sometimes will not be possible.

e) *Agents must know where they are situated, not in an absolute but rather in a relative, social, way*, such as a grid of neighbors.

Crew Intelligence Systems should divide the work of exploring and rescuing digital objects among several types of agents with different sets of capabilities, perception, and communications skills, and performs social mapping while acting. When a team of agents, each very light and with limited perception capabilities, is tasked to explore for digital objects and believe they have detected someone, they start shouting for help. This is very helpful, indeed. They do not necessarily know where they are and they are not necessarily able to return, so they just shout around, hoping that other agents will arrive to help. This avoids the strict requirements of making plans, and is the origin of the shout and act principle: agents put emphasis on action, with simple reactive behavior.

Thus S&A will be used by agents to move around, without a map, in an unknown environment of unknown complexity, while at the same time performing preservation tasks. In fact, the map is the grid of agents placed throughout the environment, the file system, the intranet and the internet. Therefore, agents must be designed to work together and to be social, autonomous, heterogeneous, self-aware, and numerous

3.2 Definition

The *Shout and Act* system is a quadruple *(E, T, A, F)* where:

1. E is a (physical) environment. The state of the environment at each moment is determined by a set of state variables VE. One of those variables is the time. S is the set of all possible states of the environment E, that is, the set of all the possible simultaneous instantiations of the set of state variables VE.
2. $T = \{t_1, t_2, ..., t_M\}$ is a set of tasks to be accomplished by the agents within the environment E. Each task t_i has associated:
   - $pre(t_i)$. A set of preconditions over VE that determine whether task t_i can be done (*introspection* and *heterogeneity*)
   - $imp(t_i)$. A nonnegative real value that reflects the importance of task t_i. (*proactivity* and *autonomy*)
   - $urg(t_i)$. A function of VE that represents the urgency of task t_i in the current state of the environment E. It will usually be a nondecreasing function of time (*proactivity* and *autonomy*)
3. $A = \{a_1, a_2, ..., a_N\}$ is a set of agents situated in the environment E. Each agent a_i may have different problem-dependent properties (e.g., weight, speed, location, response time, maximum load, perception, and communication skills, all part of the *heterogeneity*). For each agent a_i and each task t_j, $cost(a_i, t_j)$ reflects the cost for agent a_i to execute task t_j in the current state of the environment E. This cost can be divided into two parts: first, the cost for a_i to make the environment fulfill the preconditions of task t_i (this can include the cost of stopping his current task), and then, the cost for a_i to actually execute t_j. If a agent a_i is unable to adapt the environment to the preconditions of task t_j or if it is unable to carry out the task by itself, then $cost(a_i, t_j)$ is defined ∞.
4. $F : S \times A \times T \rightarrow \Re$ is the function reflecting the degree to which agents are "attracted" by tasks. Given a state s of the environment, an agent a_i, and a task t_j, $F(s, a_i, t_j)$ must be defined so that it increases with $imp(t_j)$ and $urg(t_j)$ and decreases with $cost(a_i, t_j)$

In Shout and Act, agents operate concurrently in the environment in an asynchronous manner, eliminating, thus, the typical operation cycles of other Swarm Intelligence systems (Ant Systems, Particle Swarm Optimization Systems, Cellular Automata, etc.).

The crucial step in the algorithm below is the choice of the task that the agent will execute for the next timestep. In its simplest form, this can consist of choosing the task that maximizes the attraction function F. It can also involve some kind of negotiation between agents and even some kind of local planning. One of the tasks will be to *shout* for help. The individual behavior of agents is given by the following algorithm:

Algorithm: Individual agent's behavior

```
procedure ShoutAndActAgent
  repeat
      Find the most attractive free task M
      if the agent is doing M OR trying to fulfill pre(M) then
         Continue doing it
      else
          Stop doing the current task, if any
          if pre(M) holds then
              Start doing M
          else
Do some action in order to fulfill pre(M)
end if
      end if
  until no tasks left
end procedure
```

It is worth stressing that the algorithm allows the agents to respond in real time to changes in the environment like the appearance of new urgent tasks or the temporal impossibility of fulfilling the set of preconditions of a given task.

3.3 Inter-agent Communication

S&A requires simple communicative skills in the agents to attain the coordinated and self-organized behavior typical of Swarm Intelligence Systems. We can identify three main purposes that communication can serve in order to increase S&A problem solving capabilities:

- *Conflict resolution and negotiation*. The way we defined S&A makes unavoidable the occurrence of conflicting situations in which two or more agents choose the same task to carry out. Lack of communication will lead to a waste of resources because of several agents trying to fulfill the preconditions of the same task, even to the extent that only one of them may carry it out. In such situations, it would be convenient to have some kind of negotiation method, which can be as simple as "the first one to see it goes for it". In the case study in section IV, we will discuss more elaborate negotiation strategies.
- *Perception augmentation*. For those agents with limited perception capabilities (we refer to *capability to perceive the tasks* as, for example, whether they have little skill for appraisal of the need of an object to be preserved or not), communication can allow a agent to transmit information to the others about pending tasks they are not aware of, by means of shouts and pheromones by semantic tags. Let us suppose we want to do some kind of exploratory task in a vast file sytem, which, following with the analogy, is a terrain where points of interest must attract and be explored by agents. It would be useful if agents had the ability to share information about the points of interest that they have located during their exploratory activity. In this way, agents would have access to information about the location of points of interest that lie beyond their perceptual capabilities. If agents

have the capability to sense where the shout comes from, the rule is simple: go towards the *shout*.

- *Learning*. The attraction function F does not need to be fixed in advance. Agents can learn it through their own activity and their communicative interactions with other agents. For example, a agent can discover that a certain kind of task has a high cost and communicate this fact to other agents. Furthermore, agents can even learn from other agent's ways of carrying out new tasks.

On the other hand, it is worth differentiating between two main classes of inter-agent communicative processes:

- *Direct*. Agents establish direct communication with each other via some channel and following some kind of protocol.
- *Indirect*. Agents communicate with each other through their actions and shouts, causing changes in the environment. In the S&A framework, this can be seen as Agents generating "communicative tasks" that, when carried out by other agents, increase the information they possess (about the environment and the task set). This is the case for Ant Systems, and this is also the case for Crew Intelligence Systems and for S&A.

3.4 Coalitions of Agents

Given N agents, the number of coalitions that can be generated is 2^N. Each coalition may have a value that represents how efficiently it can perform tasks [13]]. This may be due to differences in capabilities or constraints. As a matter of fact, S&A will apply to a coalition as well as a single agent. The important thing for S&A is that there are diverse behaviors and capabilities that emerge for best adapting to *E*.

4. EXPERIMENTS

Fig 1, shows a rescue scenario with a number of digital objects (victims) v_i at unknown positions in an unknown 2D map representation of a file system. We can see a hierarchical file system like a tridimensional normal distribution, where the root of the file system is in the center of the tridimensional normal. There are more than 100,000 files in more than 15,000 directories.

Also we can superpose a grid, with the center square in the normal distribution center. The value of the normal shows us a heuristic of the connections under this square. It also helps to see a matrix like a file system with links the folders to their brothers, having a similar connectivity.

The structure of the file system is projected onto a grid to give intuitive idea of how agents move and locate the victims, the digital objects to preserve, as shown in Fig. 2.

There are several agents of type A and agents of type B. The two types of agents are designed to detect and confirm the locations of victims. Also we can superpose a grid, with the center square in the normal distribution center. The value of the normal shows us a heuristic of the connections under this square. It also helps to see a matrix like a file system with links the folders to their brothers, having a similar connectivity.

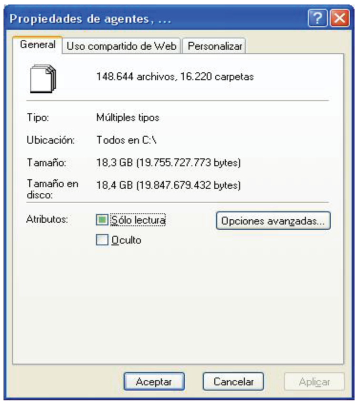

Figure 1. A file system to be made of 2DMap of Victims (objects to be preserved) and Agents

The structure of the file system is projected onto a grid to give intuitive idea of how agents move and locate the victims, the digital objects to preserve, as shown in Fig. 2.

There are several agents of type A and agents of type B. The two types of agents are designed to detect and confirm the locations of victims.

Repast (REcursive Porous Agent Simulation Toolkit http://repast.sourceforge.net/) [12] was chosen for developing the experiments, as shown in Fig 2. It is an open source freely-available agent-based simulation toolkit specifically designed for social science applications. Repast permits the systematic study of complex system behaviors through controlled and replicable computational experiments. It has been released in four versions supporting model development in three different programming languages: (1) RepastJ (Java based); (2) RepastPy (based on the Python scripting language); (3) Repast.Net (implemented in C#, but any .NET language can be used); and (4) RepastS (Repast Simphony, Java based). We use RepastS.

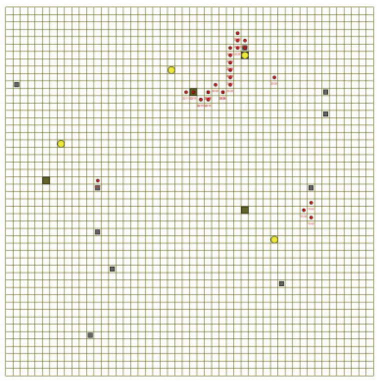

Figure 2. The radar screen in the grid of REPAST-Simphony. Victims are yellow, agents are squares, and shouts are red dots

The environment we designed in Repast Simphony consists of a 2D *radar screen* of 50 x 50 cells within an unlimited world. Agents can get lost by departing from the radar screen. In Repast, agents simulate Victims, Agents, and Shouts.

4.1 Victims

These represent the digital objects to be rescued. They are placed randomly in the grid and are static throughout the simulation. Each Victim agent is given a lifetime, with a clock that decreases with every tick of the simulation until he is rescued or dies. The lifetime of a Victim, which simulates the severity of its injuries (i.e. its format obsolescence), is one of the factors that influences the magnitude of a shout emitted by a agent that detects him, if the agent is able to determine it. Victims are shown in yellow when waiting to be rescued and green after they have been rescued. If they die, they are removed from the grid, as they might become undetectable for our agents.

So the catastrophe is a change of format for MS excel that requires.xls update to .xlsx. This update is the actual preservation action that avoids the contents of the sheet to be inaccessible in the long term.

The four digital objects are:

a) An **excel** containing a budget. High risk of losing information in the format updgrade with wrong calculations if updated.
b) A **word** document containing embedded excel objects. Moderate risk of losing information, because though it perhaps cannot update but can keep visible.
c) A **powerpoint** document containing embedded excel objects. Low risk, because it can keep the information of the excel object visible.
d) A **.rtf excel** with a budget. Very low risk, because the .rtf is designed (!) to be transportable between excel formats.

4.2 Agents

Their goal is to rescue victims or help to do so. There are 2 different types, A and B, which have different capabilities and are initially placed randomly in the grid.

- *A Type* agents are faster and more numerous than B type agents. Their main goal is to detect Victims. In this experiment, they spend most of their time moving randomly in the grid. They can perceive Victims by fast appraisal methods, and they spawn a Shout in that case. If, after a certain period of time, an A Type agent has not been able to perceive any Victim agents, it moves to the closest or strongest Shout agent emitted by another agent.

- *B Type* agents are less numerous and slower than the A Type agents, though they have superior ability to detect, appraise and rescue Victims. They follow the Shouts that A Type agents emit. When a B Type agent hears one or more Shouts, it will set the position of the Shout of highest magnitude as the target endpoint, provided that it can find a trajectory. In order to find out if this condition is met, it carries out some *introspection* and then communicates (if possible) with the other agents of its type. Thus, it determines its state (free / assisting a victim / moving to the position of a shout) and distance to the shout, and asks other Type B agents how close they are from the position where the Shout has been emitted. The next Type B agent does the same reasoning as well, deciding if it should call another agent or act by itself. With this, we achieve some basic coordination between the agents, avoiding the situation where two or more agents of equal type are assisting the same victim. Once a Type B agent reaches the position of a Shout, it performs a scan to locate and confirm the expected Victim.

4.3 Shouts

Shouts are emitted by A Type agents whenever they perceive a Victim agent. Its magnitude could be proportional to the severity of the injuries of the Victim agent. Shouts disappear some time after being emitted, and disperse with distance. The more A Type agents are shouting and the stronger the shout, the more likely it is that other B Type or A Type agents will arrive, consequently increasing the probability of detecting and rescuing a Victim.

Shouts are implemented by tagging the document, in a sort of pheromone, to help other agents detect and start appraising. They also can give pheromones in directories of several higher levels to help attract the mates. The instrument to write down all the shouts is a blackboard [14]].

5. RESULTS

First, a series of ten runs was carried out to test the stability of the experiments. Then, a series of experiments was performed with increasing complexity, different number of victims, and different number of agents of any type. Finally, three scenarios were tried, including new agents with the highest performance.

5.1 Preliminary Experiments

We run S&A 10 times, with random initial grid positions of victims and agents (controlled initial conditions), 4 victims, 10 A type agents, and 4 B type agents. The number of ticks (the discrete measure of time) it takes to rescue all the victims is recorded. We then run the algorithm with 10 B agents (we rename them C type to distinguish them from the first experiment) that move randomly like A Type agents in the same 10 initial conditions, and then compare the results. Fig 3 shows that the homogeneous agents performed slower in all cases; the Y axis represents the number of ticks needed to rescue the victims alive, and the X axis is the number of runs.

As we can see, the performance when using combined A and B type agents is higher than using only C type agents, even though C type agents have the same performance as type B agents and there is a larger number of them (10 against 4). The fact that C agents are not assisted by the faster but unreliable and low-performance A type agents makes them less efficient.

Further experiments show that to achieve similar performance, at least 18 C type agents are needed. Taking into account the fact that A type agents, due to their lower performance (in defective capacity both to detect victims and to assist them), are assumed to be of much lower cost than B/C agents, combining A and B type agents appears more suitable than increasing the number of C type agents.

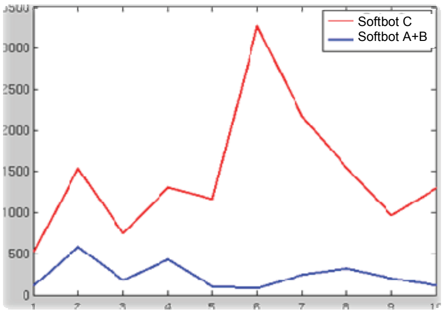

Figure 3. Preliminary experiments. 10 runs with random initial positions of 4 victims, 10 A agents, 4 B agents, and 10 C agents. Y axis is the time to rescue the 4 digital objects.

5.2 Heterogeneous Teams of A+B Agents vs. Homogeneous Teams of C Agents: Scalability Analysis

Let us see now how they behave with increasing complexity, represented as a function of the number of victims to rescue and the number of agents to coordinate. From Table 1, one can tell that with low complexity (1 to 4 victims and 1 to 4 agents), the team of C agents outperforms the heterogeneous A+B agents. On the other hand, the more complex the rescue problem (larger number of victims or agents), the more time a team of homogeneous agents needs, and consequently the worse it performs.

For the sake of simplicity of analysis, we compared the cases at three levels of complexity: low, medium, and high. This shows clearly how the growing complexity (in terms of more victims) causes the solution with C type agents to also increase in time, while the A and B types remain unchanged. In another analysis, increasing the number of agents while increasing the complexity (number of victims) causes A+B agents to rescue digital objects in a much shorter time, even with a small number of A+B agents, a trend much stronger than shown by the C agents.

The main conclusion is: in the case of *uncertain complexity* and an unknown number of available agents, the heterogeneous agents normally outperform the homogeneous agents.

Another interesting result is that 8-10 A and 4 B type agents outperform 14 B type agents. This assertion has a drawback: the B agents really need of A agents to work, so one can claim this comparison is not fair. The following section will address this issue.

Number of Robots of Type A (4 Type B robots)

Number of Victims	14	13	12	11	10	9	8	7	6	5	4
1	300	220	670	235	340	230	270	650	4090	6700	2500
2	155	190	790	890	490	390	1300	100	1500	690	650
3	420	220	120	530	110	840	170	930	1000	500	5070
4	415	170	125	45	270	370	520	1500	320	1600	630
5	120	590	270	395	180	290	130	1270	1050	530	4030
6	150	210	330	140	410	580	310	2200	2100	2100	3050
7	370	195	240	95	90	240	220	700	800	2600	900
8	1150	75	130	175	95	740	310	970	710	3200	1800
9	350	230	95	1450	850	550	110	1200	200	1300	490

High		Medium		Low	
Avrg	315	Avrg	340	Avrg	2522
Dev.	330	Dev.	382	Dev.	2256
Median	230	Median	310	Median	1500

Number of Robots of Type C (no Type A robots)

Numberof of Victims	14	13	12	11	10	9	8	7	6	5	4
1	70	840	450	210	800	127	95	157	200	1400	630
2	280	420	1050	350	230	890	450	620	450	2500	3200
3	650	450	1100	180	780	290	560	1490	3290	3000	2670
4	1200	470	1150	530	1020	1160	1970	750	2900	1600	1350
5	780	650	330	510	570	1200	1300	1070	3270	800	5200
6	970	1050	970	1800	1000	1500	2700	2050	1630	950	4900
7	680	1100	1100	950	750	2000	3200	6390	3300	2570	3700
8	1500	1350	1570	1700	790	1400	2400	4320	5300	1300	9300
9	1670	2400	530	4500	1600	2300	1780	2640	3900	4680	8790

High		Medium		Low	
Avrg	1322	Avrg	1380	Avrg	1927
Dev.	561	Dev.	625	Dev.	1257
Median	1350	Median	1200	Median	2500

Table 1. Behavior of Heterogeneous A+B and Homogeneous C type agents with different complexities of the Rescue problem

5.3 Creating Perfect Agents and the Impact on Design

There remains the question: *What if we make a team with the best properties of A and B agents, the D type super-agents?* The answer is that the mixture of A and B agents have the analogous performance as the same number of D agents. Let us examine table 2 for the features of the agents.

Case	N A	N B	N D	V A	V B	V D	P A	P B	P D	Cost (k€)
1: A+B	10	4	0	2	1	-	5	1	-	25,00
2: C	0	14		-	-	1	-	-	1	70,00
3: D	0	0	14	-	-	2	-	-	5	700,00

Type	A	B	D
Features	6	8	9
Cost (k€)	0,50	5,00	50,00

Table 2. Complexity of Case 1 Heterogeneous A+B Agents, and Cases 2 and 3 Homogeneous C and D Agents. N X means number of agents of type X, V X means the speed of the agents of type X, and P X means the perception of the agents of type **X**

	Case	Mean	Std. Deviation
Time (ticks)	1: A+B	531,37	364,87
	2: C	859,27	622,54
	3: D	432,36	274,27
Rescued	1: A+B	9,79	0,37
	2: C	9,63	0,53
	3: D	9,82	0,32

Table 3. Descriptive statistics of the experiments

The experiments are shown in Table 3, 50 runs of every case, through a complexity range of 1-9 victims, identical initial conditions in every case. The result is that the same number of A+B type heterogeneous agents perform similarly to D type homogeneous agents, even though the D type agents combine the best features of types A and B, and consequently are the most expensive. The shorter time to save digital objects implies that more digital objects are rescued.

The conclusion that the heterogeneous agent teams perform reasonably as well as the best team of homogeneous agents will have an important impact on the design of teams of agents, because, increasing the number (by 3) of A agents will perform as well as 14 D agents, that is, slight cost increments of the heterogeneous teams imply big increase of performance.

This result impacts on the design of teams of digital preservation agents: a proper mix of skilled agents with many lower skill agents should work as well though with much lower cost than the best team with only skilled powerful and expensive agents.

6. CONCLUSIONS

Creating new types of agents specifically to work cooperatively will greatly improve the efficiency of preserving digital objects, by following the analogy of rescuing people in unknown environments. This approach copes with the exponentially increasing complexity of the digital objects preservation. The fact that heterogeneous agents outperform homogeneous ones was stated in [11]] and is now shown again using algorithms like Shout and Act (S&A) that take advantage of the finding. Finally some engineering principles should be taken into account in designing Digital Objects Preservation teams, since lower-cost heterogeneous agents can achieve the same efficiency as more costly homogeneous teams.

7. ACKNOWLEDGMENTS

This research was partially funded by the European Project Num. 216746 PReservation Organizations using Tools in AGent Environments (PROTAGE), FP7-2007- ICT Challenge 4: Digital libraries and content, the PR2007-0432 Salvador de Madariaga Grant.

8. REFERENCES

[1] Dorigo, M. and Stützle, T. 2004. Ant Colony Optimization, MIT Press.

[2] Parsopoulos, K.E. and Vrahatis, M.N., 2002. Recent approaches to global optimization problems through particle swarm optimization, Neural Computing, 1 (2-3), 235–306.

[3] Holland, E. O., Beckers, E., R. and Deneubourg J.L., 1994. From Local Actions to Global Tasks: Stigmergy in Collective Robotics, 181–189, Artificial Life IV, MIT Press, Cambridge, MA.

[4] Bonabeau, E., Dorigo, M. and Théraulaz, G. 1999. Swarm Intelligence. From Natura to Artificial Systems, Oxford University Press, 1st edn.

[5] Bonabeau, E. and Théraulaz, G. 2000. 'Swarm smarts', Scientific American, March 2000, 72–79.

[6] Engelbrecht P. A.2006. Fundamentals in Computational Swarm Intelligence, John Wiley and Sons.

[7] Lewis, M. A. and Bekey, G. A. 1992. The Behavioral Self-Organization of Nano agents Using Local Rules, Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Agents and Systems.

[8] Resnick Turtles, M. 1997. Termites and Traffic Jams, Explorations in Massively Parallel Microworlds, MIT Press.

[9] Acebo E., de la Rosa J.L., Bar Systems. 2006. A Class of Optimization Algorithms for Reactive Multi-Agent Systems in Real Time Environments, 17th European Conf. on AI. (ECAI2006) Intl Workshop on New Trends in Real Time AI., pp:128-133, Riva de Garda, Italy, Aug 28-29.

[10] Quisbert, Hugo. 2008. On Long-term Digital Information Preservation Systems – a Framework and Characteristics for Development. Doctoral Thesis. 2008:77. Luleå University of Technology.

[11] de la Rosa J.Ll., Muñoz I. 2007. Learning and Adaptation in Physical Heterogeneous Teams of Agents, VIII Workshop in Physical Agents WAF 2007, pp: 9 – 18.

[12] Tesfatsion L., Iowa State University http://www.econ.iastate.edu/tesfatsi/repastsg.htm

[13] Rahwan, T. , Ramchurn, S. D. , Dang, V. D., Giovannucci, A. and Jennings, N. R. 2007. Anytime optimal coalition structure generation. In Proc. 22nd Conf. on Artificial Intelligence (AAAI), pages 1184–1190.

[14] de la Rosa, J. L., Bengtsson, J., Ruusalepp, R, Hägerfors, A, and Quisbert, H. 2008. Using Agents for Long-Term Digital Reservation the PROTAGE Project, Book Series Advances in Soft Computing Publisher - International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008), Vol. 50/2009 pp:118-p:122, ISSN 1615-3871(Print) 1860-0794 (Online), Springer Berlin / Heidelberg.

[15] Dong, J., Chen, S., Jeng, J.J., 2005. Event-Based Blackboard Architecture for Multi-Agent Systems, itcc, pp.379-384, International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II.

An Ant-like Task Allocation Model for a Swarm of Heterogeneous Robots

Sifat Momen & Amanda J.C. Sharkey

Neurocomputing and Robotics Group

Department of Computer Science

University of Sheffield,

Regent Court, Portobello, Sheffield S1 4DP, UK

Email: acq07sm@sheffield.ac.uk, amanda@dcs.shef.ac.uk

Abstract. This paper addresses the issue of applying decentralised task allocation and task switching mechanisms in heterogeneous groups of robots in order to increase their ability to respond to task demand effectively. Our work is strongly inspired by the behaviour of eusocial insects (typically ants) and their behaviour of switching tasks in order to meet the changing demand. The objective of this paper is threefold: 1) identification of task allocation and task switching mechanisms in robots inspired by ants like red harvester ants, *Pogonomyrmex barbatas*, 2) developing a simple model of these mechanisms for use with a heterogeneous group of simulated robots and 3) implementing a decentralised and adaptive mechanism of updating thresholds for heterogeneous groups of simulated robots. The paper extends the use of threshold based mechanisms in homogeneous robots into the realms of heterogeneous groups of robots. Experimental results show that the incorporation of task switching mechanisms in specialised groups of robots improves the foraging efficiency and swarm energy significantly.

Keywords: collective behaviour, division of labour, foraging efficiency, heterogeneous robots, self-organisation, task switching

1. INTRODUCTION

In recent times, there has been an increasing interest in the study of swarm robotics (SR) amongst researchers in areas as different as biology and engineering. The term *swarm robotics* was first coined in 1989 by Beni and Wang in order to describe a class of cellular robots (for brief histories see [1], [27]). Erol Şahin, in 2004, defined swarm robotics as "*the study of how a large number of relatively simple physically embodied agents/robots can be designed such that a desired collective behaviour emerges through local interaction among agents and between the agents and the environment*" [2]. The concept of SR is strongly inspired by biology especially by the observation of social insects which stand as fascinating examples of how a large number of simple individuals interact with neighbouring individuals and the environment in the vicinity to create collectively intelligent systems. The concept of SR is closely related to swarm intelligence (SI) [3] in the sense that both were initially inspired by the behaviour of social insects [4]. The field has attracted considerable interest both because it offers a number of advantages for practical applications (including robustness, flexibility and scalability) and also because of its biological inspiration.

Thus far, in much of the work in the field of SR, the robots are treated as identically able to carry out simple tasks using local communication between themselves and the nearby environment (see e.g. collective box pushing [5], cooperative transport of prey items [6], construction of a nest site [7] and foraging of food items from the environment [8, 9]). However, we argue that as SR begins to be applied in situations that are more complex in terms of the number of tasks they involve, there will be a need for more specialised heterogeneous groups of robots in order to better manage the tasks. Research in social insects has shown that social insects such as ants and bees manage complicated tasks by dividing the whole task among groups of individuals, a technique sometimes referred to as *division of labour* (DOL) [25, 26].

This paper addresses the issue of division of labour (DOL) in swarm robotics by developing a model of different groups of simulated robots/agents and exploring the ways in which their ability to carry out their individual tasks results in the accomplishment of the global task. The aim of this paper is to investigate how heterogeneous groups of robots can respond to task demand effectively. We pose two main research questions in this paper: (1) will a threshold based approach (known to work well with homogeneous groups of robots) work with heterogeneous groups of robots? and (2) does a task switching mechanism in heterogeneous groups of robots improve the foraging efficiency and the net energy gained by the swarm?

The rest of this paper is organised as follows: Section 1 introduces the concept of division of labour both in the realms of social insects and swarm robotics and describes the sources of our inspiration. Section 2 gives a description of the issues to be explored in this paper and also presents a model followed by a report of experimental investigations and results in Section 3. Finally, we conclude the paper in Section 4 with a remark on our future work.

2. Division of Labour (DOL):

Eusocial insects such as ants, bees and termites are known to be capable of carrying out different tasks concurrently. This phenomenon is what is termed "*division of labour*" [10, 11, 12, 13]. Similar mechanisms are also found within the swarm robotics literature where groups of robots are able to manage tasks as required.

2.1 Division of Labour in social insects:

DOL among social insects workers is a salient feature of their organisation. It is also fundamental to their ecological success [14, 15]. Social insects are known to divide their tasks amongst groups of workers (called castes) to improve task efficiency. In addition, one of the intriguing features of their behaviour is that they are adaptive to environmental situations and able to react to changing demand by switching tasks to those for which the colony demand at that instant is high. This feature of adaptability accounts for much of their colony success [14]. Our work is highly inspired by ants such as *Pogonomyrmex barbatas* which provide a good example of task switching in social insects.

P.barbatas ants are seed eating ants [16] and are typically between 5 mm and 7 mm long. They are widely found in the south-eastern desert of Arizona, close to the New Mexican border. These ants are known to have four different castes (patrollers, foragers, nest maintenance workers and midden workers), each having a particular task. Patrollers are one of the first groups of workers to emerge in the morning. They do most of the trail laying and also assess whether it is safe to forage or not. The successful returns of the patrollers trigger the foragers to emerge out of the nest. Foragers use the direction chosen by the patrollers and ignore food sources that were not explored by the patrollers. The nest maintenance workers are involved in building and maintaining nest chambers inside the nest while the midden workers accumulate the refuse pile or midden and move it from one place to another. Further detail of their activities can be found in [16].

Red harvester ants not only carry out their own tasks but also to change to a different task if required (Figure 1). This changing ability comes into action in response to increased demand for a particular task. For instance, when there is a flood and the nest gets damaged, the need for more nest maintenance workers would become high. In such circumstances ants from inside the nest task switch and start working as nest maintenance workers [16]. However it is worth noting that not all task-switching transitions are possible. For instance, if there is a need for foragers, all the other castes can switch their tasks to foraging and assist the foragers but if there is a need for patrollers, only nest maintenance workers are found to switch their task to patrolling.

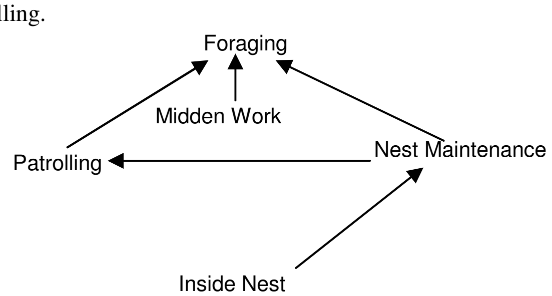

Figure 1. Task Switching in red harvester ants

2.2 Division of labour in Swarm Robotics:

In the past eight years, the field of SR has witnessed a number of developments in the areas of task allocation and division of labour. One of the earlier papers in this area was by Krieger and Billeter [8] in which a group of up to twelve mobile robots made autonomous decisions about whether to forage or rest depending on the nest energy level (which was periodically echoed to the robots in the nest from a central control station). Following this work, a number of authors have developed various ways to automatically adjust the ratio of foragers to resters. For instance, Labella, in 2004, introduced a variable delta learning algorithm [9, 17) to automatically adjust the ratio of robots undertaking the two tasks. His model was inspired by Deneubourg's learning model [18] which was developed to explain the foraging patterns observed in the *Pachycondyla apicalis* ants. Wenguo Liu and his colleagues at the Bristol Robotics Laboratory, UWE used a threshold based approach [19] to develop an adaptive threshold based mechanism for automatically adjusting the ratio of foragers to resters [20, 21] with the intention of optimising the net energy. This approach is similar to that of Labella [9,17] in the sense that the robots decide autonomously which tasks to perform via some learning mechanisms.

However, these previous studies have assumed that the robots or the simulated robots are identical (i.e. homogeneous) and are therefore able to carry out all the tasks with equal ease. We argue that this is feasible only when the complexity of the tasks is low and the number of robots involved is also low. When the complexity of the task and the number of robots involved in the problem increases, the need to divide the task among groups of robots becomes more prominent. The need for heterogeneous robot systems to tackle difficult missions has also been highlighted by various researchers in different contexts of swarm robotics (see e.g. [22, 23, 24, 9]). Numerous such examples can be found within the context of eusocial insects and one was described in section 1.1.2. With this view in mind, this paper focuses on modelling a heterogeneous group of specialised robots able to carry out their own tasks as well as switch tasks as required. In this paper, heterogeneous groups are modelled by giving a group very different threshold values compared to other groups. For instance, foragers are given a low threshold value for foraging which means that this group of robots would respond more sharply to the need of foraging compared to other task demands.

In this paper, an investigation is undertaken with the aim of exploring how task allocation and task switching behaviour in groups of heterogeneous robots affects the foraging efficiency and the net energy gained by the swarm. We conjecture that both foraging efficiency and net energy will be improved by the introduction of a task switching mechanism as this improves the flexibility of the robots' response.

3. The Model:

An agent based model (ABM) has been developed to investigate the two questions posed at the beginning of this paper (i.e. to establish whether the threshold-based approach developed in the context of homogenous groups of robots will also work with heterogeneous groups, and whether a task switching mechanism will improve their foraging efficiency.

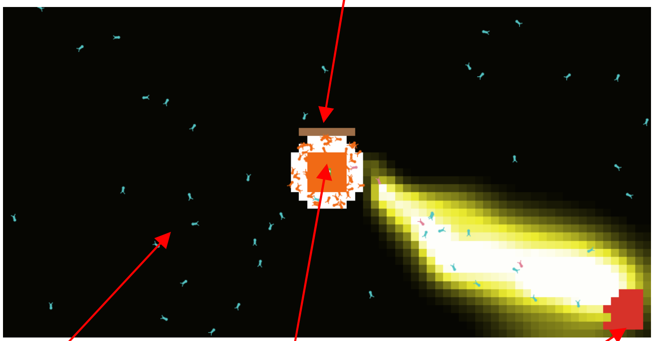

Fig 2: Snapshot of the simulation

The model places three types of agents (larvae, brood-carers and foragers) within a 40 × 20 2D environment, with the nest at the centre of the environment. The location and the tasks of the three types of castes are described as follows:

1. Larvae reside at the centre of the nest and their main task is to consume food found by foragers.

2. Brood-carers surround the larvae (and reside in the white zone) and are responsible for feeding the larvae when they are hungry.

3. Foragers, on the other hand, usually stay outside the nest and their main job is to retrieve food from the environment.

Similar spatial distributions of workers have been found in various ant species [14]. In the simulation, food appears (automatically when depleted) alternately in a clump at one of the two corners (top right and bottom right) of the environment.

3.1 Behavioural rules:

Each agent follows simple behavioural rules described as follows:

A.Larvae: Each larva possesses one of two states (hungry and non-hungry) and maintains a clock called *"hunger clock"* which is randomly initialised between 0 and 100 to ensure that all the larvae do not get hungry at the same time. The simulation starts with all the larvae in the non-hungry state. The rules followed by the larvae are as follows:

- After every time step, increment the *hunger clock* value by 1
- If the *hunger clock* value exceeds some threshold parameter specified for larvae, then switch to the hungry state
- If a larva is in a hungry state, it radiates signals (broadcast) to catch the attention of the brood carers
- If a hungry larva is fed, then switch to the non-hungry state and reset the *hunger clock* to 0

B.Foragers: Foragers can either *forage* (i.e. collect food items from the environment and bring them back to the dump area of the nest) or *rest* inside the nest. However, they have a higher propensity for foraging than resting. This is implemented by means of a threshold based approach initialised with low t_f (threshold for foraging) and high t_r (threshold for resting). At every simulation step, foragers update their threshold values as described in Table I.

Each forager also maintains two types of clocks: a *searching clock* (indicating the amount of time spent searching outside the nest) and a *resting clock* (indicating the length of time resting). Foragers follow the following behavioural rules:

- Increment the *searching clock* while searching for food
- Increment the *resting clock* if resting inside the nest
- When food is detected, take one food item and carry it back to the dump area of the nest and while going back drop pheromones on patches
- When the dump area is detected unload the food item there, rotate 180^0, set *searching clock* to zero and start a random search for the food
- If pheromone is detected when looking for food, follow the pheromone concentration to the food source
- Update the threshold values at each step of the simulation according to Table I and use the threshold values to decide whether to forage or rest.

C .Brood-carers: Brood carers can either rest or brood care (i.e. feed the larvae). They usually rest within the white zone of the nest (figure 2) and simply react to the needs of larvae by feeding them when required. This is implemented by initialising them to have high t_{bc} (threshold for brood caring) and low t_r (threshold for resting) and allowing them to update the threshold values depending on the stimulus received. Each brood carer follows the following behavioural rules:

- if it senses the *hunger-signal* of the larvae, it increases its chance of feeding the larvae by reducing its t_{bc} and increasing t_r.

When it decides to feed the larvae, it goes to the dump area, collects the food, approaches the larvae and feeds one of the hungry larvae

	Threshold for foraging (t_f)	Threshold for resting (t_r)	Reason
Successful in carrying food item back to the nest	Reduce	Increase	If the forager is successful in carrying food items back to the nest, it might mean that there are more food items in the environment to forage. The successful return of the food items thus triggers an increased chance of foraging and reduced chance of resting.
Searching clock > max-searching-clock-allowed	Increase	Reduce	If the forager spends too long searching for food, it might mean that there are not sufficient food items in the environment to forage. The foragers should therefore increase their likelihood of resting and reduce their chance of foraging.
Resting clock > max-resting-clock-allowed	Decrease	Increase	If the forager spends too long resting inside the nest, it might mean that they were resting too long inside the nest and their might already be some food available outside

Table I: Adaptation rules for foragers

4. Experiment

In order to test the two questions posed in this paper, the experiment was carried out in two phases. In the first phase the agents follow the behavioural rules outlined in section 2. In the second phase, a task switching mechanism is introduced. Brood carers, in this phase, also take part in the foraging activity if the food in the dump area falls below some threshold value. For the second phase, brood carers follow an extra behavioural rule stated as follows:

When in the dump area, check how much food is left. If the food available in the dump area is below some threshold value, then reduce the threshold for foraging (equation 1)

$$t_f = t_f - \Delta \qquad (1)$$

where Δ indicates the motivation level for brood carers to switch tasks to foraging.

As the agents move around in the environment, they can gain or lose energy. They gain energy when they are successful in carrying out their respective tasks (such as successful retrieval of food for foragers, successful feeding of larvae for brood carers and successful consumption of food for brood/larvae) and lose energy when doing some work. Table II indicates the energy gained or lost by different types of agents ('+' indicates the gain of energy while '-'indicates the loss of energy):

Activities	Energy gained/lost	Agent that loses or gains energy	Remarks
Foragers looking for food	-2 per 10 unit steps	Foragers	Energy is consumed while searching for food
Foragers resting	-1 per unit 10 steps	Foragers	Less energy is lost
Foragers carrying food	-4 per unit 10 steps	Foragers	More energy is consumed to carry the food
Successful retrieval of food by foragers	+200	Foragers	Energy gained due to the successful retrieval of food
Brood carer carrying the food from the dump area to feed the larvae	-4 per unit 10 steps	Brood carer	Energy lost in carrying the food from dump area to the centre of the nest
Successful in feeding the larvae	+3	Brood carer	Brood carers gain energy in being able to feed the larvae properly

Hungry brood	-4 per unit 10 steps	Brood	When the brood is hungry they tend to lose energy
Brood fed	+3	Brood	When a hungry brood consumes food, it gains energy

Table II: Energy expended/gained for various activities

The net energy gained by the swarm is then calculated using equation 2.

$$Net \quad Energy = \sum_{i=1}^{n_f} E_{f_i,1500} + \sum_{j=1}^{n_{bc}} E_{bc_j,1500} + \sum_{k=1}^{n_b} E_{b_k,1500} \qquad (2)$$

where n_f, n_{bc} and n_b are the number of foragers, brood carers and brood respectively and $E_{f_i,1500}$, $E_{bc_j,1500}$ and $E_{b_k,1500}$ are the individual energy of the forager, brood carer and brood respectively at 1500th simulation time steps.

4.1 First phase of the experiment:

Experiment trials	Total amount of food Collected	Net energy gained	Foragers' energy	Brood energy	Brood carers' energy
1	77	2016.4	1912	34.2	70.2
2	73	1118	1016	31.8	70.2
3	71	1108.1	954.9	62.2	91
4	80	2521.7	2397.3	43.8	80.6
5	117	9195.5	9085.5	37.2	72.8
6	79	2573.4	2381.8	82.4	109.2
7	127	10777.1	10625.1	61	91
8	118	9557.6	9336.8	98.6	122.2
9	94	4990.5	4882.5	35.2	72.8
10	142	13975.4	13779	84.6	111.8
11	83	2938.5	2817.3	43.2	78
12	72	1159	1063.8	30.2	65
13	106	6825.8	6678.2	56.6	91
14	72	740.2	651.4	26.4	62.4
Average	93.65	4964.09	4827.26	51.96	84.87

Table III: Results for Phase 1 of the Experiment

The experiment was carried out 14 times each lasting for 1500 simulation steps. The results of experiment are depicted in table III.

4.2 Second Phase of the experiment:

The second phase of the experiment was also carried out 14 times each lasting for 1500 simulation steps for various values of Δ (the motivation level of brood carers to switch their tasks to foraging). The overall result is depicted in table IV and the corresponding graphical displays in Fig 3 and Fig 4.

Δ	Average amount of food Collected	Average amount of food collected by brood carers	Average amount of food collected by foragers
0 (from first phase)	93.65	0	93.65
0.1	93	1.9	91.1
0.3	96.9	4.8	92.1
0.5	110.9	7.4	103.5
0.7	99.9	7.6	92.3
0.9	112	9.9	102.1

Table IV: Results for Phase 2 of the Experiment

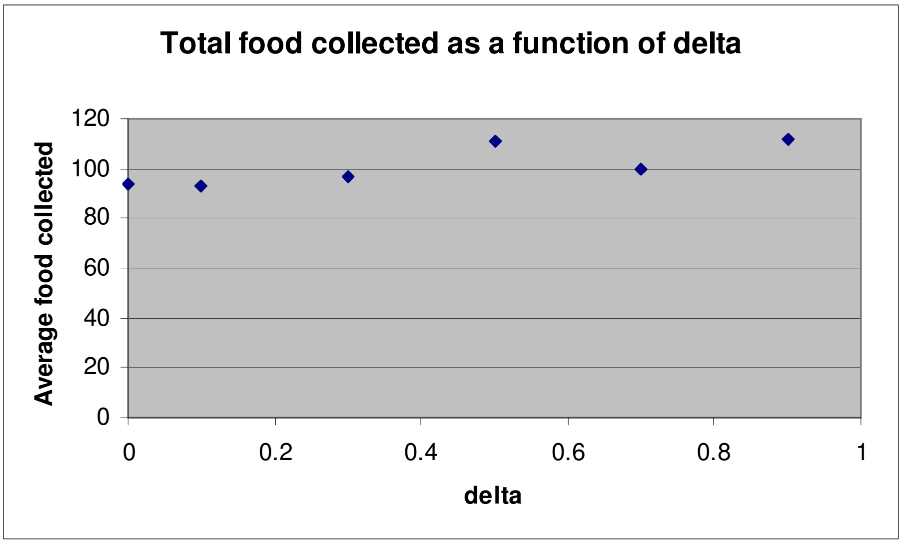

Figure 3: Total amount of food collected as a function of delta

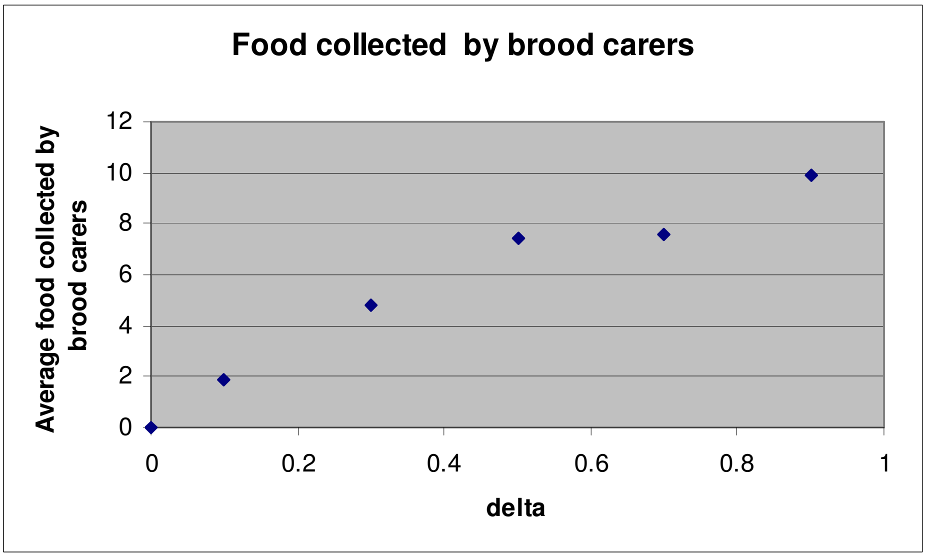

Figure 4: Food collected by brood carers as a function of delta

Figure 4 shows that the average amount of food collected by brood carers has a linear relationship with Δ (product moment correlation coefficient = 0.97). Regression analysis reveals the relationship between the food collected by brood carers (f_{bc}) and Δ as

$$f_{bc} = 0.89 + 10.51\Delta \quad (3)$$

Figure 5 shows that the incorporation of task switching improves the swarm energy by more than double than in the case of no task switching. Furthermore, results in Table IV indicate that the incorporation of task switching improves the overall foraging efficiency (the total amount of food collected).

Such mechanisms would be highly beneficial in the context of heterogeneous robot systems where specialised robots are grouped to carry out group-specific tasks. Both their efficiencies and swarm energy can be improved by employing task switching behaviours between the groups.

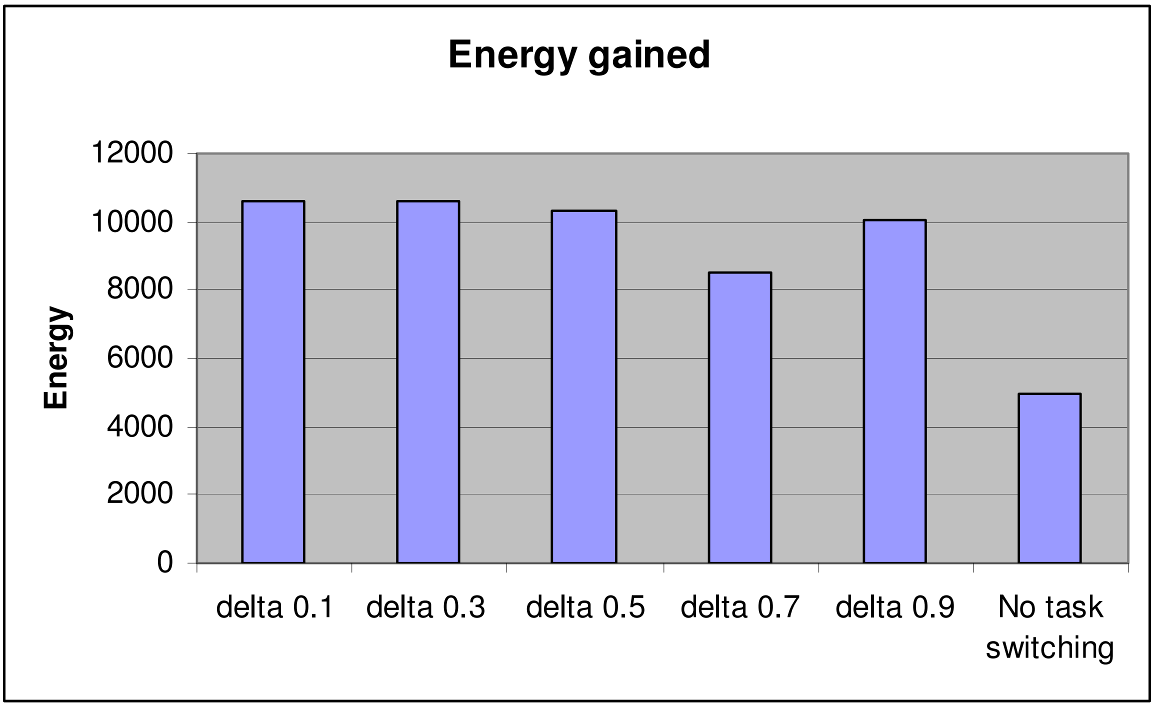

Figure 5: Net energy gained by the swarm

5. Conclusion:

We have presented a task allocation and task switching model for heterogeneous groups of autonomous mobile robots/agents that adhere to the swarm robotics' principles of local control and local communication. We have shown that the foraging efficiency and the net energy gained by the swarm are affected by the degree of task switching. A threshold based approach has been embraced for the task switching mechanisms and an adaptive mechanism introduced for updating the thresholds of the simulated robots. The model was inspired by the behaviour of social insects, in particular that of the red harvester ants *Pogonomyrmex barbatas*. Simulation results demonstrate that agents are able to perform tasks adaptively, flexibly and efficiently which suggests that the threshold based mechanism works well in the heterogeneous environment as well. It further shows that introduction of the task switching mechanism improves both the foraging efficiency and the net energy gained by the swarm significantly. Such a result would be highly beneficial in the case of heterogeneous groups of robots carrying out more than one task concurrently.

Our future work will concentrate on investigating how the distribution of food in the environment affects the foraging efficiency and the net energy gained by the swarm. We also plan to develop and empirically evaluate the effectiveness of different task switching algorithms in the context of heterogeneous robot systems.

REFERENCES

[1] Beni, G., "From Swarm Intelligence to Swarm Robotics", In Swarm Robotics, (Eds) Erol Şahin and William M. Spears, LNCS 3342, pp: 1 – 9, (2004)

[2] Şahin, E., "From sources of inspiration to domains of application", In Swarm Robotics, (Eds) Erol Şahin and William M. Spears, LNCS 3342, pp: 10 – 20, (2004)

[3] Beni, G., Wang, J., "Swarm Intelligence in Cellular Robotic Systems", Proceeding of NATO Advanced, Workshop on Robots and Biological System, (1989)

[4] Sharkey, A.J.C., "Robots, Insects and Swarm Intelligence", Artificial Intelligence Review, vol: 26, pp: 255 – 268, (2006)

[5] Kube, C.R., Zhang, H., "The use of perceptual cues in multi-robot box pushing", IEEE International Conference on Robotics and Automation, pp: 2085 – 2090, (1996)

[6] Kube, C.R., Bonabeau, E., "Cooperative transport by ants and robots", Robotics and Autonomous Systems, vol: 30, pp: 85 – 101, (2000)

[7] Parker, C.A.C., Zhang, H., Kube, C.R., "Blind Bulldozing: multiple robot nest construction", Proceedings of the 2003 IEEE/RSJ Conference on Intelligent Robots and Systems, pp: 2010 – 2015, (2003)

[8] Krieger, M.J.B., Billeter, J.B., "The call of duty: Self organised task allocation in a population of up to twelve mobile robots", Robotics and Autonomous Systems, vol: 30, pp: 65 – 84, (2000)

[9] Labella, T.H., Dorigo, M., Deneubourg, J.L., "Efficiency and task allocation in prey retrieval", ed. A.J. Ijspeert et.al, BioADIT 2004, LNCS 3141, pp: 274 – 289, (2004)

[10] Oster, G.F., Wilson, E.O., "Castes and ecology in social insects", Princeton University Press, Princeton, (1978)

[11] Robinson, G.E., "Regulation of division of labour in insect societies", Annual Review Entomology, vol: 37, pp: 637 – 665, (1992)

[12] Bonabeau, E., Dorigo, M., Theraulaz, G., "Swarm Intelligence: From natural to artificial systems", Oxford University Press, (1999)

[13] Hölldobler, B., Wilson, E.O., "The Ants", The Belknap Press of Harvard University Press Cambridge, Massachusetts London, England, (1990)

[14] Bourke, A.F.G., Franks, N.R., "Social evolution in ants", Princeton University Press, (1995)

[15] Franks, N.R., "Ants", Encyclopaedia of Insects, ed. V.H. Resh and R.T. Carde, Academic Press, pp: 29 – 32, (2003)

[16] Gordon, D.M., "Ants at work: how an insect society is organized", The Free Press, (1999)

[17] Labella, T.H., "Division of labour in groups of robots", PhD thesis, Universite Libre de Bruxelles, (2007)

[18] Deneubourg, J.L., Goss, S., Pasteels, J.M., Fresneau, D., Lachaud, J.P., "Self-organization in ant societies (III): Learning in foraging and division of labour", From individual to collective behaviour in social insects, (Eds) J.M. Pasteels and J.L. Deneubourg, Experientia Supplementum, vol: 54, pp: 177 – 196, (1987)

[19] Bonabeau, E., Theraulaz, G., Deneubourg, J.L., "Quantitative study of fixed threshold model for the regulation of division of labour in insect societies", Proceeding of the Royal Society of London B, vol: 263, pp: 1565 – 1569, (1996)

[20] Liu, W., Winfield, A.F.T., Sa, J., Chen, J., Dou, L., "Towards energy optimisation: Emergent task allocation in a swarm of foraging robots", Adaptive Behaviour, vol: 15, no: 3, pp: 289 – 305, (2007)

[21] Liu, W., Winfield, A.F.T., Sa, J., Chen, J., Dou, L., "Strategies for energy optimisation in a swarm of foraging robots", ed. E. Şahin, W.M. Spears and A.F.T. Winfield, Swarm Robotics 2006, LNCS 4433, pp: 14 – 26, (2006)

[22] Momen, S., Amavasai, B.P., Siddique, N.H., "Mixed species flocking for heterogeneous robotic swarms", IEEE Eurocon 2007, The international conference on 'computer as a tool', pp: 2329 – 2336, (2007)

[23] Momen, S., Sharkey, A.J.C., "An ant-like task allocation model for heterogeneous groups of robots", 4th European Meeting of IUSSI 2008, pp: 117, (2008)

[24] McLurkin, J.D., "Analysis and implementation of distributed algorithms for multi-robot systems", PhD thesis, MIT, (2008)

[25] Jeanson, R., Fewell, J.F., Gorelick, R., Bertam, S.M., "Emergence of increased division of labour as a function of group size", Behav Ecol Sociobiol, vol: 62, pp: 289 – 298, (2007)

[26] Anderson, C., McShea, D.W., "Individual *versus* social complexity, with particular reference to ant colonies", Biol. Rev., vol: 76, pp: 211 – 237, (2001)

[27] Sharkey, A.J.C., "Swarm robotics and minimalism", Connection Science, vol: 19, no: 3, pp: 245 – 260 (2007)

AquaRobots Phase II: Oil Spillage Detection Using Swarm AquaBots

Babu K Shamanna [1] and **Aladdin Ayesh** [2]

Abstract. Applications of underwater and space exploration robots and those operating in hazardous environments can benefit from the use of multi-robot system. In challenging applications like these, multi-robot systems can deal with wide range of tasks, cope with environmental changes including the loss of robots, faster, and often relatively cheaper. The main aim of the project reported in this paper is to build autonomous marine robots (AquaBots), to deploy these AquaBots in group to form a swarm and assign them the task of searching the surrounding area for any oil spillage. In the prototypes, that spillage is simulated using target material. Finally optimize the number of robots in a swarm based on the time estimates to achieve this goal.

Keywords *- Aquarobots, swarms, search robots, spatial detection.*

1 Introdcution

Swarm robots is the branch of robotics which deals with the study of how a large number of relatively simple agents can be designed and coordinated such that a desired collective behaviour emerges from their interactions among the agents and between the agents and the environment [11]. It was initially inspired by the emergent behaviours observed in insects, ants, termites, wasps and bees which stand as fascinating examples of how a large number of simple individuals can collectively interact to create intelligent systems. Social insects are well known to coordinate their actions to accomplish tasks that are beyond the capabilities of a single individual; for example termites build large and complex mounds, army of ants organize impressive foraging raids, and ants can collectively carry large preys [1].

Swarm robotics is one of the active areas of robotics research with enormous recent advances in this field [2]. This paper on swarm robotics presents the way in which how swarm intelligence can be implemented to address one of the major challenges that the world is facing today and demonstrates how this can be resolved.

Nowadays oil spills in oceans presents the potential for enormous harm to deep Ocean and coastal fishing and fisheries [3]. Therefore the aim of the project is to build water surface robots (Aquabots) extending on previous work in developing small scale aquarobots [10], and to deploy them in group to form a swarm and assign them the task of searching the surrounding area for any oil spill or exotic material. Finally optimize the swarm number by changing the team size for a give task i.e. to estimate the appropriate team size required to achieve this goal by measuring the time taken.

[1] De Montfort University, Leicester LE1 9BH, UK, email: babu22blr@gmail.com

[2] Intelligent Mobile Robots and Creative Computing Research Group, De Montfort University, Leicester LE1 9BH, UK, email: aayesh@dmu.ac.uk

2 BACKGROUND

Swarm robots find its theoretical roots in studies of animal societies such as bees and ants. Social insects are valuable source of inspiration for designing collectively intelligent systems comprising many agents [4].

2.1 Problem Domain

Oil pollution arising either from marine accidents or from routine ship operations is one of the major problems that threaten the equilibrium of the marine environment. They produce both economical and ecological damage. Wildlife including fish, sea creatures, mammals, reptiles, amphibians and birds that live in or near the ocean are also poisoned by oil waste. Oil waste that invades and pollutes these areas and negatively affects human activities can have devastating and long-term effects on the local economy and society [3].

The efforts in protecting the environment after an oil spill (through an anti-spill operation) could cost billions of dollars in cleanup and damage costs (sources ITOPF). Therefore marine oil pollution has attracted increasing research effort over past few decades to overcome this problem [5]. So, It is important for us to understand how overwhelm of modernisation has impacted the ecosystem. Today oil spills in water bodies is one of the major issues that we have to look upon acutely and take necessary measures to save the marine life.

In order to demonstrate the capabilities and potential of swarm robots to tackle this problem AquaBots where developed and tested in a pool of area 2.62 m2. AquaBots searched the surrounding area to locate the target/exotic material by avoiding any obstacle in their path. In each of the iteration the complexity of the problem (number of obstacles) was changed in order to check robustness of the control algorithm implemented. The experiments were carried out under certain test conditions like the robots were tested in an indoor environment since the sensors were calibrated for these conditions. Treating AquaBots as mobile sensors is the subject of another ongoing research [12].

2.2 Methodologies adopted in overcoming this Problem.

The Elimination Unit for Marine Oil Pollution (EU-MOP) project uses this multi-robot system to respond to oil spills in oceans [6]. According [6] the operation of the working of the multi robots system is divided into different phases, after the detection of oil and initial phase, the robots and support vessel the so called mother ship will be ready for cleaning operation. Later the robots will start the oil recovery operation with the help of skimmers. As every unit has only limited energy supply and limited storage capacities onboard, they have to move back to the mother ship in the case its energy is running low or its oil storage is full. As soon as the whole oil spill is eliminated the units will move back to the mother ship. In another similar project by [7] uses autonomous multi-robot system for confinement of spilled oil. In this case, a simulation study was made, initially with two ships towing the boom and later on extended to swarm of robots which navigate through and confine the oil spill. But the robots are not capable of removing spills; they would only confine the spills by not allowing the oil to spread. In the project presented in this paper the ultimate focus was locating the object in a confined search area and identify whether the object is target material or not. When the target material was found the AquaBot stops immediately. In other words, our focus is the speed and consistency of time needed in finding the spillage spots in the infected area.

3 GENERALISED MODEL OF SYSTEM

While the aquabots perform the search operation, the interaction between the aquabots or between the aquabots and environment is reactive behaviour. The control is decentralised i.e. the aquabots are not in network. The advantage of having this type of controller is that any number of aquabots can be appended to the team at any point of time without re-synchronising the entire system [8].

The whole search operation of aquabot is based on set of predefined rules. So, the aquabots just follows the rules based on their priorities. The aquabots during the search operation should be capable avoiding obstacles, if any, in their path. The threshold value for the boats to change their direction in case of obstacle was determined by experimentation and ultrasonic range sensor was used for this purpose.

The Aquabots consisted of light sensor pointing downward as shown in figure1. Two ultrasonic sensors and one infrared sensor were used for changing the direction in case any obstacle in their path. RCX microcontrollers consisting of three input ports to read the sensors value and three output ports for the actuators were used. Lego motors with internal gearing were used to propel the boats [9].

The light sensor continuously searches the target object based on the light intensity. If the target object is found the boats stop immediately and helps in locating the target substance but if the object doesnt match the template (light intensity), the search operation is continued. The infra-red sensor and ultrasonic sensors mounted at the front measures the distance from nearest obstacle and compares it with threshold value, if the distance is less than the threshold value the boat changes the direction or it continues to move in straight path. The boat turns right if left ultrasonic senor reading is greater or turns left if right ultrasonic sensor reading is greater. This process is looped until the target object is found.

The Aquabots were implemented in swarm, initially with one aquabot and gradually the number was increased to four to determine the appropriate size of team required to locate the target object in a test area of 2.62 m2. It was found that the search time decreased as the team size was increased but after certain critical value the time again increased due to interference between the aquabots. The experiments revealed that three aquabots were appropriate and resulted in minimal search time.

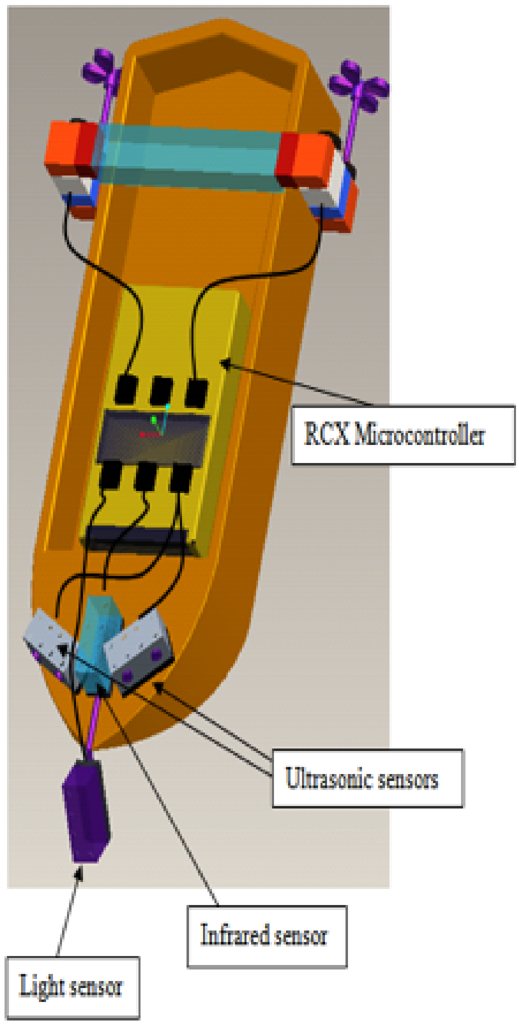

Figure 1. CAD model of Aquabot

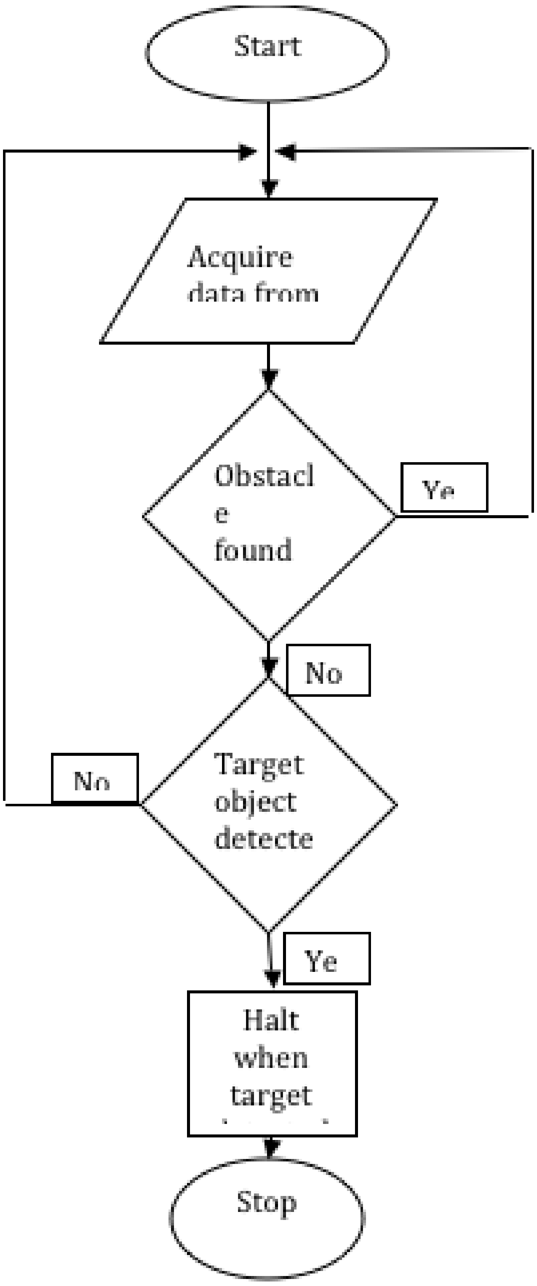

Figure 2. Flow chart of Aquabot control

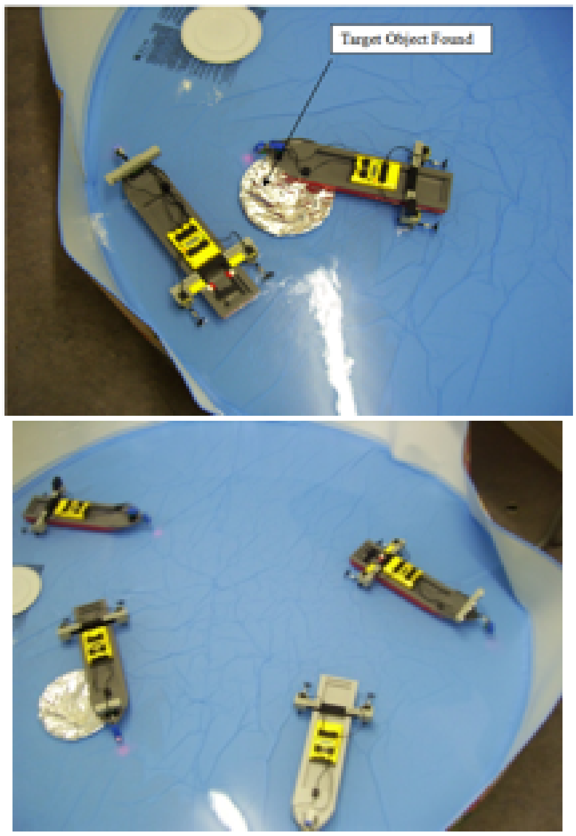

Figure 3. Robots Swarm searching for the target object in teams of two and four respectively.

4 Experiment and Analysis of Results

Several experiments were conducted to estimate the time required to locate the target object and the results are tabulated as shown below [3]. Initially the experiment was conducted with one Aquabot to locate the target object.

Experiment No.	No. of Robots	Time Required to Locate Target (mins)
1	1	3.12
2	1	2.29
3	1	2.54
4	1	3.24
5	1	2.01

Table 1. Time taken for one AquaBot to search the target material

From table 1 we can calculate the average time using one AquaBot is 2.642 minutes and the standard deviation was found to be 0.475.The experiment was repeated by varying the team size in order to determine optimum number of AquaBots required to achieve this task at best consistent time duration. The timings are presented in table 2.

Experiment No.	No. of Robots	TRLT (mins)	Standard deviation
1	1	2.64	0.475
2	2	1.54	0.286
3	3	2.21	0.229
4	4	2.09	0.263

Table 2. Time taken for one AquaBot to search the target material

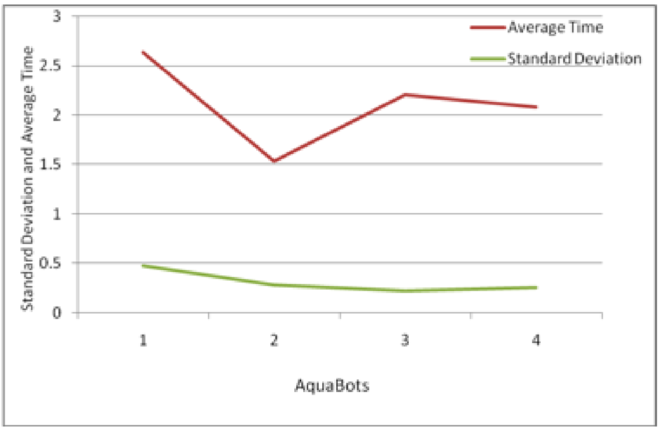

Figure 4. Variation of time and standard deviation in relation to AquaBots number

From Table 2 Standard deviation seems to decrease greatly once a second AquaBot is introduced but the difference in standard deviation between second and fourth AquaBot is small.

From the above experiments it is clearly evident that the average time taken to achieve the goal decreases when the team size increases to two but the time taken again increases. This is because when the team size is larger than that is required the AquaBots interfere with each other and slow down the search operation. Another critical issue that was noted during the experiment was, when the boats located the target object they moved a little from their location due to inertia of the boat even after the motors were turned off and there was collision among the members at times due to inefficiency of the sensors.

[3] Video clips from the experiments are available at: http://www.youtube.com/aayesh15

5 Conclusion

Taking all constrains into account, the outcomes of the project was reasonably good. The project clearly demonstrated the development process of Aquabots and their implementation in swarm to accomplish the task of search operation. The Aquabots were capable of manoeuvring independently and all the aim and objectives mention in this project were achieved. Several experiments were conducted to demonstrate the capabilities of Aquabots to locate the target object by avoiding the obstacles in their path. When the Aquabots were implemented in Swarm to optimize the group size, it was found that three Aquabots were appropriate and resulted in minimal time to locate the target object for a test area of 2.62 m2.

Since Lego mindstorm robotic kit was used in development of Aquabots, it resulted in some of the limitations like the inability of the microcontroller to take more than three inputs and give out only three outputs. The absence of Bluetooth module in microcontroller hindered the ability of members of swarm to communicate effectively. The use of vision based camera could result in better object recognition capabilities of Aquabots.

During the search operation, the sensors used in Aquabots for obstacle avoidance like ultrasonic sensors were not fully efficient because when the transmitter of ultrasonic sensor sends signal and if the reflecting surface is not even the Aquabots could not detect the presence of obstacle as result they collided with the obstacle.

Since the inertia of the Aquabot was the hindrance in term of achieving the dynamic stability of the system, review of mechanical design could lead to much better dynamic stability of the Aquabots. This can be achieved by changing the positions of motors and other components in the system.

Reference

[1] Erol sahin William M. Spears and Alan F.T. Winfield Workshop on Swarm Robotics SAB06 Rome, Italy, Sep 30 and Oct01, 2006.

[2] Bonabeau, E.; Dorigo, M.; Theraulaz, G(1999): Swarm Intelligence: From Natural to Artificial Systems. New York, Oxford. Oxford University Press, 1999.

[3] Water Encyclopedia 2007 [online] available from http://www.waterencyclopedia.com/Oc-Po/Oil-Spills-Impact-on-the-Ocean.html [Accessed on 3/6/2008]

[4] Marco Dorigo, Elio Tuci1, Roderich Gro, Vito Trianni, Thomas Halva Labella, Shervin Nouyan, Christos Ampatzis, Jean-Louis Deneubourg, Gianluca Baldassarre, Stefano Nolfi, Francesco Mondada, Dario Floreano, and Luca Maria Gambardella (2005): Swarm Robotics The SWARM-BOTS Project Volume 3342/2005, ISBN 978-3-540-24296-3.

[5] ITOPF (The International Tanker Owners Pollution Federation Limited) [online] available from http://www.itopf.com/spill-compensation/cost-of-spills/ accessed on 2/09/2008.

[6] Psaraftis, H. N Ventikos, N. P. (2006): An intelligent robot system to respond to oil spills: the EU-MOP Project. In: Proceedings of the INTERSPILL, London

[7] Jimenez, J.F.; Giron-Sierra, J.M.; Dominguez, A.; De la Cruz, J.M.; Riola, J.M. (2005) Ships confining an oil spill over: a scenario for automatized cooperation. Issue , 20-23 June 2005 Page(s): 1226 1231, Volume 2.

[8] Batalin, A M. and Sukhatme, S M. (2002) Spreading Out: A Local Approach to Multi-robot Coverage. In Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems. pp. 373-382, Fukuoka, Japan, June 25-27, 2002.

[9] Ferrari,M., Ferrari,G., Hempel,R. (2002) Building robots with Lego Mindstorms, Rockland: Syngress

[10] M. O. Daglioz and A. Ayesh, "Aqua Swarms: Design and Implementation of Water Surface AUV," Swarm Intelligence Algorithms and Applications Symposium (SIAAS'08) - AISB 2008 Convention, Aberdeen, Scotland, 2008, pp. 38-43.

[11] G. Al-Hudhud and A. Ayesh, "Real Time Movement Coordination Technique Based on Flocking Behaviour for Multiple Mobile Robots System," Swarm Intelligence Algorithms and Applications Symposium (SIAAS'08) - AISB 2008 Convention, Aberdeen, Scotland, 2008, pp. 31-37.

[12] M. Al-Obaidy and A. Ayesh, "Optimizing Autonomous Mobile Sensors Network using PSO Algorithms," International Conference on Computer Engineering and Systems (ICCES'08), Cairo, Egypt, 2008