Population Dynamics Analysis in an Agent-based Artificial Life System for Engineering Optimization Problems

Abraham Prieto, Pilar Caamaño, Francisco Bellas, Richard J. Duro, *Senior Member, IEEE*

***Abstract*— In this paper we discuss the relevance of performing a population dynamics analysis to improve the results obtained using agent-based artificial life systems for optimization. The present study derives from our work trying to solve engineering optimization problems using a distributed approach based on agent's interactions. We have realized that a simple analysis of the population dynamics can show the relevance of some variables and energy exchange rates in the stability of the system. The results obtained can be used to control the equilibrium points and/or avoid non-convergence (population extinctions) by changing the initial conditions or the parameters of the energetic model used in the system. To illustrate the results of such population dynamics analysis, a practical example based on a routing algorithm is presented.**

I. Introduction

TYPICAL engineering problems requiring complex optimization procedures such as industrial processing lines, marine transportation routes, hydrodynamic simulations and so on are suitable for the application of agent-based search algorithms that exploit distributed computation to deal with the high dimensionality of the resulting solution spaces. In this line, in the field of Artificial Life (ALife), different approaches related to the application of agent-based systems to non-biological practical problems may be found [1]. Particularizing to function optimization, Yang et al [2], have considered distributed algorithms, which can be successfully applied to parallel optimization and design. The underlying idea is the same one we have been using: complex tasks may be performed by distributed activities over massively parallel systems made up of computationally simple elements. In his work, Yang obtained very promising results applying an emergent colonization algorithm for optimum design of short bearings [3]. Another relevant paper in engineering optimization is [4] where the authors use a hybrid of a conventional ALife algorithm together with random taboo search in the design of engine mounts.

All of these approaches use agent-based ALife models in the optimization of a target function. Our work is more focused in the emergence of optimization algorithms instead of optimizing particular functions. This idea was originally presented in Langton's Artificial Life book [5] with a chapter devoted to the design of artificial problem solvers. In this line, Hillis developed in [6] a very interesting procedure that coevolves an optimization algorithm population against a population of parasites that are continuously increasing the complexity of the search space.

In our research, we have developed and tested a methodology to solve engineering optimization problems using an agent-based ALife system [7][8] that has provided very promising results. In these works, we typically obtain as a final result the agents' parameters that define the optimization algorithm. These parameters are extracted from the genes of the agents in the final stable state reached by the system. It is obvious that such stable state depends on several system variables, some fixed by the problem to solve but some others selected by the designer in an empirical way. For example, the environment resource input rate, life cycles, energy exchange between agent and environment, etc. At this point, some questions arise: in case that our system does not reach a stable state, how could we determine whether the system is not converging – extinctions, or oscillating behavior - because the agents are not able to adapt -learning problems- or because the parameters selected make the system unstable? Is the combination of parameters we have selected forcing the system to a fixed solution in terms of number of agents and resource-related parameters? Can we modify the equilibrium point to obtain different solutions to the problem? To answer this kind of questions, we have modeled the relations between the energetic variables in the system and studied how they affect the equilibrium, this is, we have analyzed the population dynamics of the system obtaining very promising results that will be explained in section IV.

Reviewing the literature about population dynamics analysis in order to improve our approach, we have found surprisingly that in very relevant ALife subfields, like virtual ecosystems, this kind of analyses are not typically carried out. [9]. On the other hand, in more biological ALife approaches, [10][11] similar studies exist applied to the energetic models that support the system. In this subfield, such energetic models are very important and they are even considered the best-developed aspect of ecosystem ecology. In fact, the energy flows are responsible for various phenomena such as the creation of trophic levels, food chains and webs, productivities and efficiencies. Additionally, behaviors related with energy flow affect the evolutionary adaptation of an organism and as a consequence it would seem obvious that an in depth analysis of the energetic model variables that would allow them to be tuned should be very relevant for these systems.

Taking into account that, in our application, the tuning of the variables of the system and the control of the evolution is

Abraham Prieto, Pilar Caamaño, Francisco Bellas and Richard J. Duro are with the Integrated Group for Engineering Research, Universidade da Coruña, Spain, email: {abprieto, pcsobrino, fran, richard}@udc.es

even more important because the obtained parameters will be directly applied in engineering problems, this kind of analysis becomes essential.

The objective of this work is, initially, to present a generic description of the population dynamics analysis we have performed, and later to show a practical implementation of these analyses in a specific engineering optimization case: a routing problem. To make easier the understanding of this particular implementation, it is necessary to include a detailed description of all the elements involved in the problem and the results obtained. Thus, the paper is structured as follows: in section II we formally explain the population dynamics analysis that has been carried out; in section III we present in detail a practical experiment that will be used to illustrate the analysis; in section IV we show the kind of information and control of the system that can be achieved and, finally, in section V we will present some conclusions of this work.

II. Population Dynamics Analysis

The first thing we have to take into account is that, for a population dynamics analysis, we should ensure that the growth rates for one species of the ALife system are related with the density of individuals of other species, like in typical *Lotka-Volerra* or *Logistic Growth Equation* models. So, in any ALife system we need to adapt the interactions between species to ensure this inter-relation. The variation of the density of each species population ($\Delta\rho_{species\_n}$) has to be a function of the other species density:

$$\Delta\rho_{species\_n} = \varphi(\rho_{species\_n}, \rho_{species\_i}, \rho_{species\_j}, \ldots,) \quad (1)$$

In a real model, for example in the case of the simple prey/predator model, the probability of a prey finding a resource point is directly related to the prey/resource density and, consequently, when resources decrease, this density also decreases and the average path length for finding resources increases.

Now, we should establish the balances that control the energy flows of the system and the number of individuals of each species. In any ALife system we can establish two different types of energy balances. The first energy balance ($E_{environment\_energy}$) is related with the energy available in the environment on each time step through the following equations:

$$\Delta E_{env\_energy} = R_{input} - R_{species\_consumption} - R_{output} \quad (2)$$
$$R_{species\_consumption} = \sum_i R^i_{species\_consumption} \quad (3)$$

The resources input rate (R_{input}) and the resources output rate (R_{output}) are defined by the designer, but the rate of resources consumed by individuals of all the species ($R_{species\_consumption}$) depends on the number of individuals in the system (N_{ind}), the average time (T_{avg}) to reach a resource and the unitary consumption C_{unit} when reaching a resource, through the expression:

$$R_{species\_consumption} = \frac{N_{ind} C_{unit}}{T_{avg}} \quad (4)$$

In the equilibrium state, the energy variation must be zero, so this provides an equation relating all the variables involved:

$$\Delta E_{energy} = 0$$
$$R_{input} = R_{species\_consumption} + R_{output} \quad (5)$$

Secondly, we can establish the energy balance for the individuals of every species:

$$\Delta E^i_{species\_energy} = R^i_{species\_cons} - N^i_{ind} C_{time\_loss} - L^i_{task} \quad (6)$$

N^i_{ind} stands for the number of individuals of species *i* and C_{time_loss} stands for the unitary loss of energy of the agents in each time step. Note that, in this balance, $R^i_{especies\_cons}$ is now an input. Finally, L^i_{task} represents the losses of energy associated to some tasks, such as mating, fighting, etc.

Again, in the equilibrium state, the energy variation must be zero, so:

$$\Delta E^i_{species\_energy} = 0$$
$$R^i_{species\_cons} = N^i_{ind} C_{time\_loss} + L^i_{task} \quad (7)$$

Using these balance equations, we can obtain general relations between parameters for the stable state. In the next section, we present the details of a typical optimization engineering problem, a routing problem, solved using the methodology presented in [8] that will be used to show how these equations can be used in a real case (section IV).

III. Application Example

As a simple application test, we have chosen a routing problem that consists in *finding an algorithm* that provides the best route to go from an origin to a target point on a random graph. The problem was implemented using the WASPBED (World-Agent Simulation Platform for BEhavior Design) simulation environment presented in [7]. It provides the capability of defining all the elements in an ALife system, changing the environments, the definition of the participating elements or the constraints with a minimum effort and it permits developing a creation template useful for all the different configurations of the different problem environments. For this example, we have designed an ALife environment made up of the following elements (see Fig. 1):

The graphs: they contain nodes and edges linking nodes. There are two special nodes, the origin and the target. We have situated 6 random graphs with 12 nodes and 25 edges each. Each edge has a state parameter (pheromone density) associated to it that can be modified. The graph has a state parameter representing the amount of resource present in the target node that is decreased by the agents when they arrive there.

The agents: that travel around the edges of the graphs. Their relevant state parameters are their position, age, activation level and an itinerary memory. Their descriptive parameters (inheritable) are four coefficients of the control system and another one controlling pheromone production. In the simulation they are represented by circles and their size represents the age. The different colours of the circles represent the different combination of parameters in their genes.

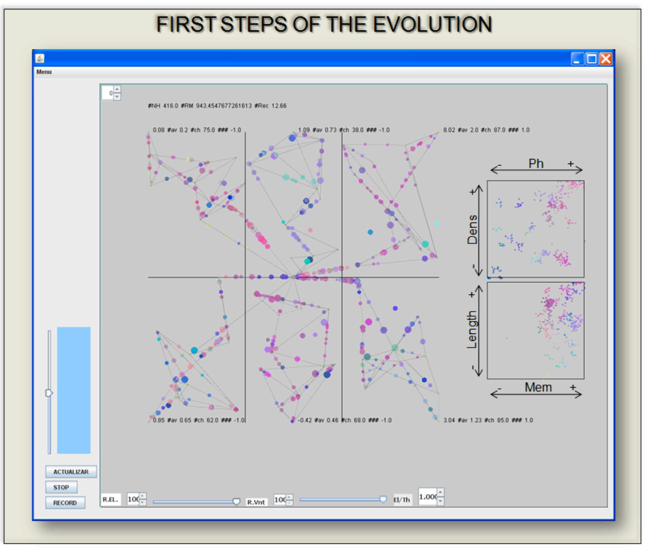

Fig. 1. View of the WASPBED simulator with the elements of the example. The agents are represented by circles (the size represents their age) and the graphs are represented by interconnected lines.

The interaction rules (events) in this case are:

Advance: each agent goes through the current edge towards the next node.

Choosing the next edge: this represents the control system. It decides the next edge using an evaluation function. It is activated when the agent is on a node.

Reproduction: when an agent reaches the origin after having reached the target, that is, after a complete cycle, it performs a crossover process of its inheritable parameters with any other agent in the same situation in order to create a new agent.

Graph regeneration: when one graph's resources run out, new random nodes and edges are generated.

Pheromone release: when an agent finds the target node, it may increase the pheromone density of the edges it goes through on its way back to the origin node, the pheromone release rate is regulated by the pheromone production parameter.

Itinerary memory update: Every time an agent leaves an edge, its memory is updated adding this new edge.

Agent death: when the agent's age exceeds its longevity value it is removed from the population.

With these rules we have defined an environment where the agents go through the different graphs leaving different pheromone trails, choosing their paths and reproducing depending on their controller parameters. As time progresses, they improve their ability to find better routes between the origin and target nodes due to the association between reproduction and finding the origin and target nodes.

The control system decides the next edge based on an evaluation function that consists on the sum of four product terms: the pheromone level of the evaluated edge times the *pheromone density coefficient*, the logarithm of the number of agents on the edge times the *agent density coefficient*, the square root of the position of the edge in the itinerary memory times the *memory coefficient* and the edge length times the *edge length coefficient*. All of the terms are normalized between 0 and 1. The best evaluated edge will be followed by that individual. Thus, different coefficients determine different behaviors for the different individuals, so the resulting collective behavior will be represented by the combination of the coefficients of the whole population. An additional parameter is used, the pheromone production rate, which represents the amount of pheromone an ant produces when it leave an edge on its way back to the origin.

As soon as we started with the simulations, we noticed that the initial definition of the rules described above presented a shortcoming: when the population improves, the agents select shorter paths and the resource consumption increases. As a consequence, the individuals tend to perform better but the system can't assure that the resources are available when the target point is reached. As a consequence the system degrades its performance.

This happens mainly because the agents do not have any sensor providing information on the resource levels in a graph and therefore they do not have information on the existence of resources until they reach the target point. In addition, eventually the graphs become too easy to solve, then the extinction of a graph occurs frequently. The main implication of this is that the success of an individual becomes more and more random and less focused on improving efficiency. To solve this problem, we have introduced in the system a predator/prey model based rule that uses a new parameter in the graphs: the advance velocity.

We have allowed the system to evolve with several different configurations changing different parameters of the environment rules. Those we consider the most interesting to analyze are the following:

We let the system evolve with the previously described rules and, after a few minutes, the population reaches a stable state. Its genetic parameters are shown in Fig. 2.

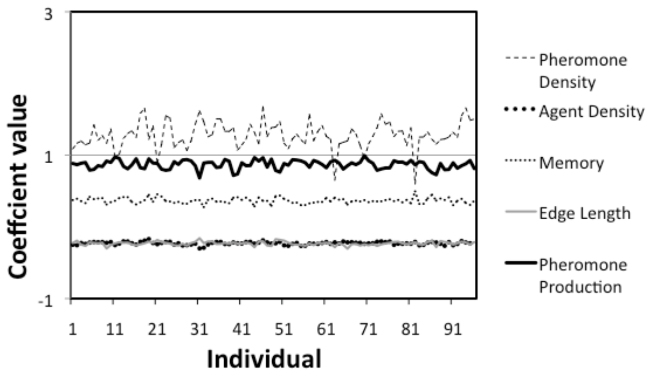

Fig. 2. Resulting algorithm parameters for the whole population using the original rules

As this figure shows, the most relevant parameters are the memory coefficient and then the pheromone density, which means that, initially, the agents select, out of the available edges, those that are more separated in its memory. That is, they select those in which they have not been before or where they have been a long time ago. Secondly, the agents select the edges that have the highest pheromone density. The agent density is used in third place with a low and negative value, but it is also of interest since, despite the fact that it is not decisive in most of the edge selections, it helps to spread population about into several graphs at the starting point or after the deactivation of a graph (Remember that all the graphs are linked and when an individual is choosing a new graph, each graph is treated as an edge with an equivalent pheromone level and agent density). Finally the edge distance coefficient has a value near zero, nevertheless this parameter is not very important because each of the randomly generated graphs could have an optimum composed by short or long edge indistinctly, in fact, on several evolutions the first three coefficients preserve the same values and the this one varies.

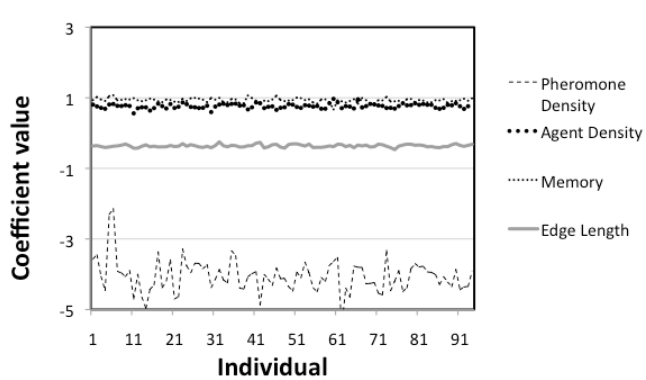

Fig. 3. Resulting algorithm parameters for the whole population without updating the pheromone values of the edges

In a second case, we let the system evolve without updating the pheromone values of the edges, that is, the communication through the environment (stigmergy) the agents were using is no longer possible. With this modification, we obtain the population parameters that are shown in Fig. 3. As we can see, the behavior is now guided by both the memory and the agent density. The individuals in this case behave in a similar way as in evolutions with pheromone updating. That is because they are choosing the most populated edges and consequently most of them are following the same paths. Thus, they are in fact using another way of indirectly communicating information and taking advantage of it. The other coefficients are not relevant: the pheromone coefficient has no effect, the pheromone value of the edges is now always zero, and the edge length is not related with the quality of the solution, as explained before.

In the last configuration we want to highlight, we have tried to make the problem a little bit harder. A maximum lifetime has been defined for the graphs, and then the graphs are deactivated. So now graphs are deactivated when they run out of resources or when they reach their time limit. Pheromone updating is allowed in this case. The first consequence of this rule is that there is almost no time to exploit a graph after finding the good paths. Thus, basically, the agents are most of the time exploring new graphs and finding solutions as fast as possible. The coefficients obtained are shown in Fig. 4 and correspond to a behavior that is not very different from the first we analyzed except for one aspect: the pheromone coefficient is now twice as high as in the first case. The memory coefficient and the pheromone production levels are the same. This means that the agents assign a higher relevance to the pheromone trail. This can be considered normal as now there is no time for the pheromones level to increase too much.

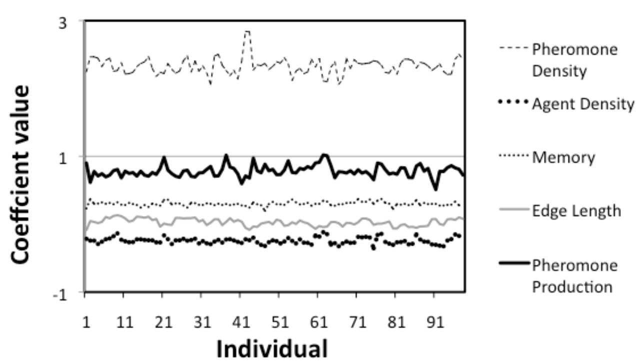

Fig. 4. Resulting algorithm parameters for the whole population with a maximum lifetime for the graphs

Thus, to conclude this section, we must point out that after 2000-3000 iterations, which implies 4-6 minutes CPU time with a 2 Ghz Intel Core 2 Duo, the simulations converge to a dominant type of agents, this is, most of the agents have very similar coefficients. We have represented in Fig. 5 one of the stable populations as shown in the WASPBED tool. On the right part of the image, we show a representation of the four coefficients of the whole population. As we can see, most of the individuals have converged to the same parameter combination. It is important to point out that memory is fundamental, as we have seen in all the simulations with several configurations, otherwise the individuals are easily trapped in loops around the same nodes or edges. In fact, after the first time steps all those individuals that do not use their memory disappear.

Our efficiency measure is the average number of steps needed to travel between the origin and target. We are not measuring it directly, instead, we measure the density of existing resources which is directly proportional to the average number of steps. Despite fluctuations due to changes in the environmental conditions this measure decreases during evolution.

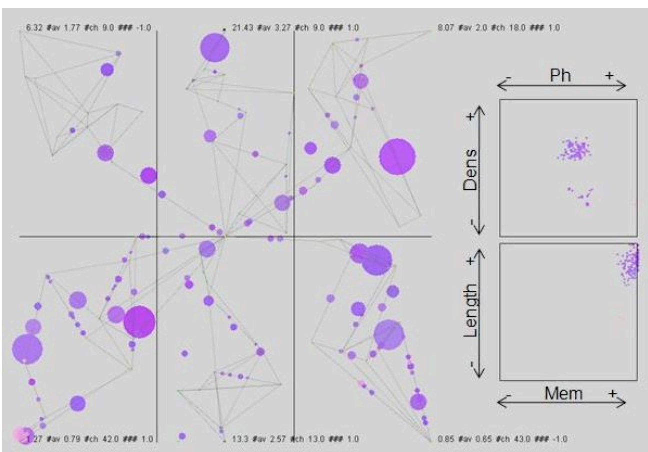

Fig. 5. Capture of the WASPBED tool with a stable population.

IV. Example Analysis

Once we have presented the details of the practical example we will use to perform the population dynamics analysis, we can particularize the equations presented in section II for this particular routing case.

In order to satisfy the condition established in equation (1), which implies that the average path required to find a resource point is directly related with the prey/resource density we start by assuming an apparent fixed surface of area (S) (representing the area of the territory in which the system evolves), that together with the available resources level (R_{lev}) will lead us to an apparent resource density (D_{ap}):

$$D_{ap} = \frac{R_{lev}}{S}$$

In this case, the average path length (L_{av}) to reach a resource will be proportional to a characteristic length and, therefore proportional to the square root of the surface area, and finally, inversely proportional to the squared root of the apparent resource density:

$$L_{av} = \frac{c_0}{\sqrt{D_{ap}}} = \frac{C}{\sqrt{R_{lev}}}$$

As a consequence, when the resources decrease, the density of the resources decreases too and the average path required to find a resource increases and the other way around.

In some scenarios, the number of resource providers is always the same and they are always in a fixed position, as for instance in the routing problem we are using for this application example. So, the density of resources is always the same.

To adapt this kind of simulation we propose that instead of modifying the scenario length as a function of the resource density, which is always an option but albeit more complicated, we modify the advance velocity so that the average time to cover a graph stretch is the same. Thus, starting from an initial length (L_0) and an initial advance velocity (v_0), the instantaneous velocity may be expressed as (v_{adv}):

$$T_{elapsed} = \frac{L_{av}}{V_0} = \frac{L_0}{v_{adv}} \Rightarrow v_{adv} = \frac{L_0 v_0}{L_{av}} = C'\sqrt{R_{lev}}$$

According to the analysis performed in section II, there are two resource balances that control the population dynamics, represented by the equations (2) and (6). The rules we have imposed in our simulation imply that:

- The available resources are increased with a constant rate
- The individuals lose energy with a unitary constant rate.
- When an individual reaches the target point, it increases its energy level and decreases the environment energy level.
- In this case, the resources output is not considered

As a consequence, in this case, the energy level (E_{i+1}) in this system can be expressed by:

$$E_{i+1} = E_i + R_{input} - R_{species\_consumption}$$

In equation (4), the term (T_{avg}) average time is obtained from the average real length (L_{avgR}) and the advance velocity:

$$T_{avg} = \frac{L_{avgR}}{v_{adv}}$$

Finally, we can obtain the average real length as the product of the average base length (L_{avgB}) and an interference factor (F_i) that changes with the number of individuals and with resource density. This factor represents the fact that sometimes one agent starts looking for a resource but in the middle of the trip and before reaching it another agent ingests the resource. This makes the average length increase. This interference factor will be always larger than or equal to one.

$$R_{species\_consumption} = \frac{N_{ind} C_{unit}}{L_{avgB} F_i} C' \sqrt{R_{lev}}$$

In the equilibrium state, as established in equation (5), the consumption rate and the input rate must be equal, so:

$$R_{input} = R_{species\_consumption} = \frac{N_{ind} C_{unit}}{L_{avgB} F_i} C' \sqrt{R_{lev}}$$

Consequently, the environment energy balance follows the expression:

$$N_{ind} = \frac{C'' L_{avgB} F_i}{\sqrt{R_{lev}}}$$

with $C'' = \frac{R_{input}}{C_{unit} C'}$

Note that, as commented in section II, R_{input} is a constant value defined by the designer. In this expression, we don't know the interference factor, which must be experimentally obtained, or the L_{avgB} coefficient, that cannot be taken as a fixed value but rather as a distribution with a mean and a standard deviation.

To obtain a curve corresponding to this expression, we have run the simulation several times without considering reproduction and death of the agents, and analyzing the stabilization value of the resource level. The following results were obtained for the previous example of route optimization using the WASPED simulation tool [8]:

R_{lev}	N_{ind}
8.27	512
8.09	362
9.12	256
15.19	181
20.92	128
32.91	90
69.77	64
141.38	45
272.72	32
495.88	22
911.54	16
4240.32	8
17121.73	4

We have represented these data in the curve shown in Fig. 6 relating the number of individuals with the available resources. We are also taking into account the standard deviation obtained for the data shown in the previous table. The area within the two curves is where 75% of the values obtained are.

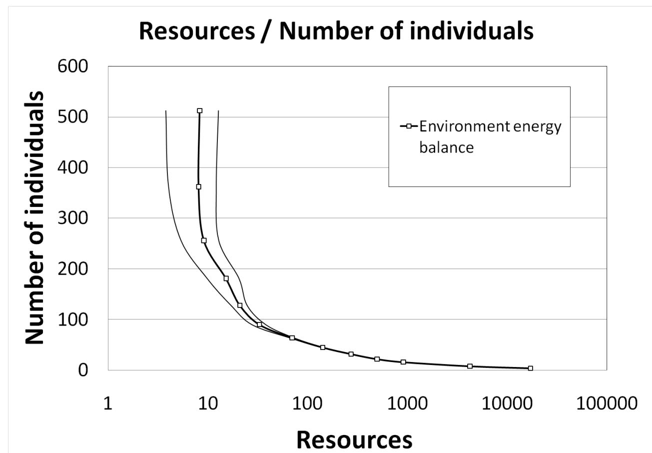

Fig. 6. Evolution of the available resources with the number of individuals using the environment energy balance

From this curve we can extract an experimental value for $L_{\exp}$ that represents the average base length multiplied by the interference factor, obtaining thus the experimental length:

$$L_{\exp} = L_{avgB} F_i = \frac{N_{ind} \sqrt{R_{lev}}}{C''}$$

In this case, the values obtained are:

N_{ind}	L_{exp}
512	8679.00
362	6066.73
256	4555.10
181	4156.92
128	3449.79
90	3042.16
64	3149.72
45	3152.43
32	3113.48
22	2886.37

16	2846.07
8	3069.21
4	3083.69

We can now establish the equation for the energy of the individuals, according to equation (6) and considering that:

- In every time step each individual is consuming one unit of energy
- There is no loss of energy associated to special tasks
- The reward for completing the graph is e_{unit}

The resulting equation is:

$$E_{i+1} = E_i - N_{ind} + \frac{N_{ind} e_{unit} C' \sqrt{R_{lev}}}{L_{\exp}}$$

When in equilibrium, as established in equation (7):

$$L_{\exp} = K\sqrt{R_{lev}}$$

$$K = e_{unit} C'$$

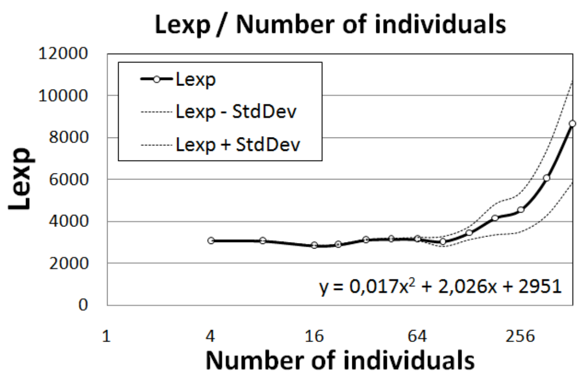

Fig. 7. Evolution of the number of individuals with the experimental length of the graphs

This leads us to an implicit relationship between the number of individuals and the available resources. The experimental data for coefficient L_{exp} can be fitted to a second order polynomial curve with high precision, which shows that the individual energetic balance, as can be seen on experimental data in fig. 7, follows an equation like:

$$N_{ind} = a + \sqrt{b + \sqrt{R_{lev}}}$$

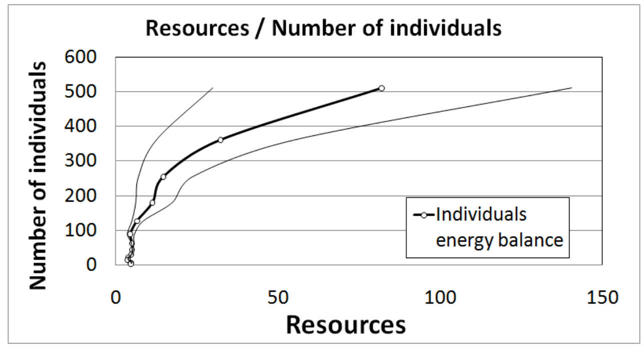

Fig. 8. Evolution of the available resources with the number of individuals using the individual energy balance

If we obtain the experimental data for this expression and represent it, we obtain the curve shown in Fig. 8. At this point, we can cross the two curves that characterize our system (represented in figures 6 and 8) and create the representation shown in Fig. 9.

As we can see in Fig. 9, the equilibrium zone of this system (where the two curves intersect) is wide, which corresponds to an unstable but non-divergent system. Variations in the quality of the population can displace both curves right or left producing an increase or decrease of the available resource density, which is a very interesting indicator of the quality of a population.

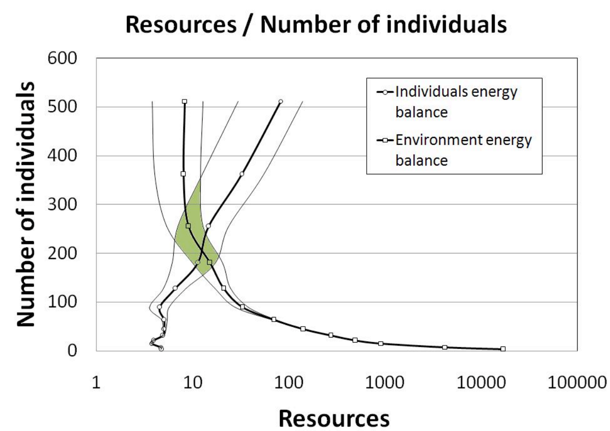

Fig. 9. Representation of the two energy balance curves that characterize our system and the stable area (dark).

As we can see from this last figure, using the equations presented in section II and the empirical curves obtained for each particular ALife system, if needed, we can, on one hand, predict general population dynamics without simulating all the possible cases and, on the other, learn how every parameter involved is affecting the equilibrium state or stability. As a consequence, we can tune the involved parameters to vary the stable area seeking some specific solutions with a given number of agents or a given rate of resources, for example. In addition, with the results of this analysis we could modify the initial conditions to guarantee or speed up convergence.

V. Conclusion

In this work we have presented a general procedure to analyze agent-based artificial life systems in order to obtain the formal relations between the main variables of the system. The procedure is based on two balance equations, one for the environment energy balance and the other one for the individual energy balance. We have shown with a typical engineering optimization problem, a simple routing example, that this kind of simple analysis permits a tuning of the variables of the system to obtain a desired stable state and, as a consequence, a desired solution. Currently, we are studying this procedure to predict the results of evolutions before they are done, with successful initial results.

Acknowledgment

This work was supported by the Ministerio de Ciencia e Innovación of Spain through project DPI2006-15346-C03-01 and DEP2006-56158-C03-02.

References

[1] Kyung-Joong Kim, Sung-Bae Cho, "A Comprehensive Overview of the Applications of Artificial Life", *Artificial Life 12* (2006) pp153-182

[2] Yang, B.S., Lee, Y.H., "Artificial Life Algorithm for Function Optimization", *Proceedings 2000 ASME Design Engineering Technical Conferences and Computer and Information in Engineering* (2000)

[3] Yang, B.S., Lee, Y.H., Choi, B.K., Kim, H.J., "Optimum design of short journal bearings by artificial life algorithms". *Tribology International*, 34 (2001) pp 427-435.

[4] Ahn, Y. K., Song, J. D., Yang, B.S. "Optimal design of engine mount using an artificial life algorithm", *Journal of Sound and Vibration*, 261 (2004) pp 309-328.

[5] Langton, C.G. "Artificial Life: an overview". *MIT Press* (1995)

[6] Hillis, W. D. "Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure", *Artificial Life II* (1992) pp 313-324

[7] Abraham Prieto, Richard Duro, and Francisco Bellas, "Obtaining Optimization Algorithms through an Evolutionary Complex System", *Abstracts booklet ECCS07*, pp.132-132, European Complex Systems Society, 2007

[8] Richard J. Duro, Abraham Prieto, Francisco Bellas, Pilar Caamaño, "A Complex Systems Based Tool for Collective Robot Behavior Emergence", *Lecture Notes in Artificial Intelligence*, vol. 5271, pp. 633-640, Springer-Verlag, (2008)

[9] Dorin, A., Korb, K. and Grimm, V., "Artificial-life ecosystems: what are they and what could they become?". *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pp. 173-180. MIT Press, Cambridge, MA (2008)

[10] Reiners W.A, "Complementary models for ecosystems", *American Naturalist*. 127:59–73, (1986)

[11] Grimm V., "Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the future?", *Ecological Modelling*, Volume 115, Number 2, 15 February 1999 , pp. 129-148(20)