

Experimental analysis of a variable size mono-population cooperative-coevolution strategy

Olivier Barrière and Evelyne Lutton

Abstract Cooperative coevolution strategies have been used with success to solve complex problems in various application domains. These techniques rely on a formulation of the problem to be solved as a cooperative task, where individuals collaborate or compete in order to collectively build a solution. Several strategies have been developed depending on the way the problem is shared into interdependent subproblems and the way coevolution occur (multipopulation versus monopopulation schemes). Here, we deal with a mono-population strategy (Parisian approach) applied to a problem related to the modeling of a cheese ripening process (french Camembert). A variable sized population Parisian GP strategy has been experimented, using adaptive deflating and inflating schemes for the population size. Experimental results show the effectiveness of the approach on real data collected on a laboratory cheese ripening production line.

1 Introduction

Cooperative co-evolution techniques mimic the ability of natural populations to build solutions via a collective cooperation process. Nowadays, these techniques are used with success on various problems [9, 25], including learning problems, see [3] for a recent reference on the topic. The large majority of these approaches (CCEA algorithms) deals with a coevolution process that happens between a fixed number of separated populations [19, 4, 21]. We study in this work a different implementation of cooperative coevolution principles, the so-called Parisian approach [6, 18], that uses cooperation mechanisms within a *single* population. It is based on a two-level representation of an optimization problem, in the sense that an individ-

INRIA Saclay - Ile-de-France
Parc Orsay Université, 4, rue Jacques Monod, 91893 ORSAY Cedex, France
{Olivier.Barriere, Evelyne.Lutton}@inria.fr
<http://apis.saclay.inria.fr>

ual of a Parisian population represents only a part of the solution to the problem. An aggregation of multiple individuals must be built in order to obtain a solution to the problem. In this way, the co-evolution of the whole population (or a major part of it) is favoured instead of the emergence of a single best individual, as in classical evolutionary schemes. The motivation is to make a more efficient use of the genetic search process, and reduce the computational expense. Successful applications of such a scheme usually rely on a lower cost evaluation of the partial solutions (i.e. the individuals of the population), while computing the full evaluation only once at each generation (see figure 1).

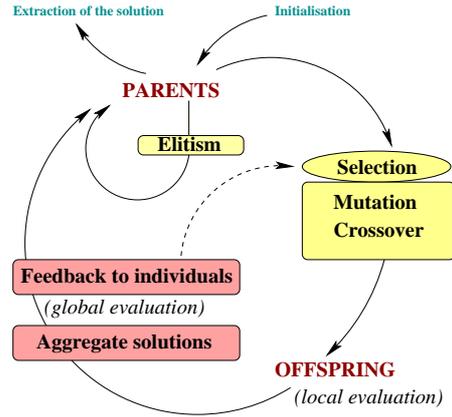


Fig. 1 A Parisian EA: a monopopulation cooperative-coevolution

In this paper, we design a variable-sized population adaptive scheme and test it on real data provided by an experimental cheese ripening process. Various population sizing and resizing schemes have been studied in the literature for classical evolutionary schemes [15, 10]. It has been clearly stated that adaptive population size allows to build more efficient optimisation algorithms, by dynamically balancing the exploration and exploitation capabilities of the search, the gain in efficiency being measured in terms of number of fitness evaluations.

Common on-line population size adjustment schemes are related to the improvement of the best individual of the population, to the variance of population fitness, or rely on the notion of age and lifetime of individuals. There also exists strategies based on competing subpopulations, for example [23] proposed a scheme based on competing subpopulations: each subpopulation is running a different search strategy, and regularly compete with each other. The size of “good” strategies then increases while “bad” ones decreases, the sum of the sizes of all population being constant.

However, to the best of our knowledge, there exists no work of this type for cooperative-coevolution schemes. The strategy we experiment in this paper for monopopulation cooperative-coevolution relies on the notion of global fitness improvement, and allows to allocate less local fitness evaluations to obtain a better result *in*

fine. Tests have been performed in order to evaluate the improvements due to population deflation, then to population deflation + inflation, in comparison to a constant population size scheme.

The paper is organised as follows: section 2 describes the experimental framework¹, and the Parisian GP implementation for phase prediction, that serves as a basis for testing the adaptive population sizing schemes we propose. This scheme is described in section 3. The experimental analysis of section 4 proves the effectiveness of the population deflation and population deflation + inflation schemes. Conclusions and future works are detailed in section 5.

2 Cooperative-coevolution learning on agrifood data

The successful use of evolutionary optimisation methods in general and genetic programming (GP) in particular for the resolution of complex problems related to agrifood has been attested by various recent publications [2, 11, 24]. In the present work, we deal with a cheese ripening process: The cheese, during ripening, is an ecosystem that is extremely complex to be modeled as a whole, and where human experts operators have a decisive role. The modifications of substrate under the action of several populations of micro-organisms is only partially known, and various macroscopic models have been experimented to embed expert knowledge, like expert systems [13], neural networks [14], mechanistic models [22], or dynamic Bayesian networks [5]. The major problem common to these techniques is related to the sparseness of available data: collecting experimental data is a long and difficult process, and resulting data sets are often uncertain or even erroneous. The precision of the resulting model is often limited by the small number of valid experimental data, and parameter estimation procedures have to deal with incomplete, sparse and uncertain data.

In a previous contribution we compared a Bayesian network model whose structure has been built using expert knowledge with evolved GP estimators. Two strategies have been considered: a “classical” GP and a cooperative/coevolutionary GP strategy, see [1]. Experimental results proved the efficacy of GP approaches to estimate the phase parameter of the process (currently made “at hand” in industrial production lines).

¹ This study is part of the large INCALIN research project, whose goal is the modeling of agrifood industrial processes. It is supported by the French ANR-PNRA fund.

2.1 *The test-problem: phase estimation of a Camembert-cheese ripening process*

For soft-mould cheese ripening, relationships between microbiological and physicochemical changes depend on environmental conditions (*e.g.* pH, temperature, relative humidity ...) [16] and influence the quality of ripened cheeses [17].

A ripening expert is able to estimate the current state of the process, and control decisions are then generally based on this subjective but robust expertise. The ripening process is usually divided in four phases:

- **Phase 1** is a drying process: the surface of cheese is initially very wet and progressively gets a rather dry aspect.
- **Phase 2** corresponds to the apparition of a *P. camembertii*-coat, *i.e.* the white-coat of Camembert.
- **Phase 3** is characterized by the thickening of the creamy under-rind.
- **Phase 4** let appear strong ammoniac smell and dark brown colour on the cheese.

The expert's knowledge is obviously not limited to these four stages. But the identification of these stages help to evaluate the whole dynamics of ripening and to detect some drift from the standard evolution.

2.2 *Phase estimation using a Parisian GP*

A Genetic Programming approach is used to search for a convenient formula that links micro-organisms proportions to the phase at each time step t (static model), without *a priori* knowledge of the phase at $t - 1$.

Instead of searching for a phase estimator as a single monolithic function, phase estimation can actually be split into four combined (and simpler) phase detection trees as shown on figure 2. The structures searched are binary output functions (or binarised functions) that characterize one of the four phases. The population is then split into four classes such that individuals of class k are good at characterizing phase k . Finally, a global solution is made of at least one individual of each class, in order to be able to classify the sample into one of the four previous phases via a voting scheme detailed at the end of this section.

2.2.1 Search space and local fitness measurements

The derivatives of four variables are considered², *i.e.* the derivative of pH (acidity), la (lactose proportion), Km and Ba (two lactic acid bacteria proportions). We search for formulas of type: $I(\frac{\partial pH}{\partial t}, \frac{\partial la}{\partial t}, \frac{\partial Km}{\partial t}, \frac{\partial Ba}{\partial t})$ with real outputs mapped to binary outputs, via a sign filtering: $(I() > 0) \rightarrow 1$ and $(I() \leq 0) \rightarrow 0$.

² It has been shown in a previous work that these parameters are relevant to estimate the phase [1, 20]

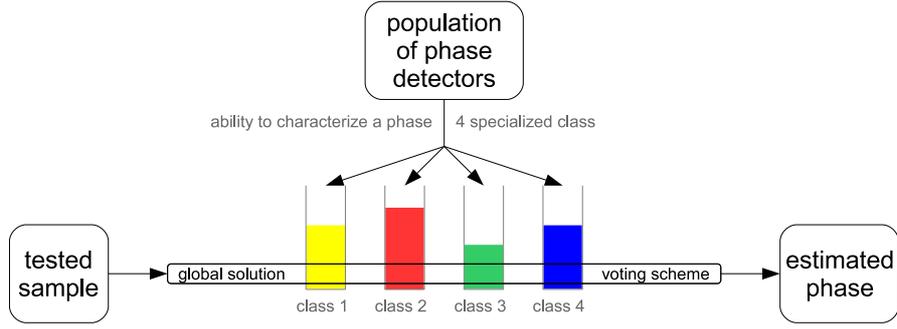


Fig. 2 Phase estimation using a Parisian GP. Four classes of phase detectors are defined: individuals of class k are good at characterizing phase k .

The function set is made of arithmetic operators: $\{+, -, *, /, \wedge, \log\}$, with protected $/$ and \log . The terminal set is made of the four partial derivatives plus real constants. The constant's values are not limited, but randomly initialised using one of the following laws $\mathcal{U} [0, 1]$, $-\mathcal{U} [0, 1]$, $\mathcal{N} (0, 1)$, also randomly chosen. (\mathcal{U} is the uniform law, and \mathcal{N} the normal law)

Using the available samples of the learning set, four values can be computed, in order to measure the capability of an individual I to characterize each phase:

$$k \in \{1, 2, 3, 4\} \quad F_k(I) = 3 \sum_{i, \text{phase}=k} \frac{I(\text{sample}(i))}{\#\text{Samples}_{\text{phase}=k}} - \sum_{i, \text{phase} \neq k} \frac{I(\text{sample}(i))}{\#\text{Samples}_{\text{phase} \neq k}}$$

i.e. if I is good for representing phase k , then $F_k(I) > 0$ and $F_{\neq k} < 0$

The local fitness value, *to be maximised*, is a combination of three factors:

$$\text{LocalFit} = \max\{F_1, F_2, F_3, F_4\} \times \frac{\#\text{Ind}}{\#\text{IndPhaseMax}} \times \frac{\text{NbMaxNodes}}{\text{NbNodes}} \Big|_{\text{if NbNodes} > \text{NbMaxNodes}}$$

The first factor is aimed at characterising if individual I is able to distinguish one of the four phases, the second factor tends to balance the individuals between the four phases ($\#\text{IndPhaseMax}$ is the number of individuals representing the phase corresponding to the *argmax* of the first factor and $\#\text{Ind}$ is the total number of different individuals in the population) and the third factor is a parsimony factor in order to avoid large structures. NbMaxNodes has been experimentally tuned (currently fixed to 15) in order to limit the size of the trees (in term of number of nodes).

2.2.2 Genetic operators

Genetic operators are classical GP operators, i.e. tree crossover via exchange of subtrees, and subtree and point mutations (see table 1). Real values are considered separately and undergo a real mutation with probability p_{rm} as a multiplicative perturbation according to a χ^2 law of parameter N , parameters vary linearly with generations, see [1].

From	to	probability
operator	operator	0.1
variable	variable	0.1
variable	constant	0.05
constant	variable	0.05
constant	constant	p_{rm} : 0.1 to 0.5 N : 1 to 1000

Table 1 Probabilities of point mutation operators

2.2.3 Sharing distance

The set of measurements $\{F_1, F_2, F_3, F_4\}$ provides a simplified representation in \mathbb{R}^4 of the discriminant capabilities of each individual. As the aim of a Parisian evolution is to evolve distinct subpopulations, each being adapted to one of the four subtasks (i.e. characterize one of the four phases), it is natural to use an euclidean distance in this four dimensional phenotype space, as a basis of a simple fitness sharing scheme [8].

2.2.4 Aggregation of partial solutions and global fitness measurement

At each generation, the population is shared in four classes corresponding to the phase each individual characterises the best (i.e. the argmax of $\max\{F_1, F_2, F_3, F_4\}$ for each individual). The 5% best of each class are used via a voting scheme to decide the phase of each tested sample³ (see figure 2). The global fitness measures the proportion of correctly classified samples on the learning set:

$$GlobalFit = \frac{\sum_{i=1}^{learning_set} CorrectEstimations}{\#Samples}$$

The global fitness is then distributed as a multiplicative bonus on the individuals who participated in the vote: $LocalFit' = LocalFit \times (GlobalFit + 0.5)^\alpha$.

As $GlobalFit \in [0, 1]$, multiplying by $(GlobalFit + 0.5) > 1$ corresponds to a bonus. The parameter α varies along generations, for the first generations (a third of the total number of generations) $\alpha = 0$ (no bonus), and then α linearly increases from 0.1 to 1, in order to help the population to focus on the four peaks of the search space.

³ This scheme may also yield a confidence level of the estimation. This measurement is not yet exploited but can be used in future developments of the method.

3 Variable size population strategies

3.1 Population size decrease scheme

3.1.1 Local fitness measurements

Several fitness measures are actually used to rate individuals, namely the raw fitness *rawfitness*, i.e. the set of four values $\{F_1, F_2, F_3, F_4\}$, that measure the ability of the individual to characterize each phase, the local fitness *localfitness* = $\max(\text{rawfitness})$ which represents the best characterised phase, and the adjusted fitness *adjfitness* = $\frac{\text{localfitness}}{\mu} \times \frac{\#IndPhaseMax}{\#Ind} \times \frac{\#NodesMax}{\#Nodes} \times \text{bonus}^\alpha$, which includes sharing, balance, parsimony and global fitness bonus terms.

3.1.2 Redundancy - Diversity's hidden iceberg

Because of the binarised output which only takes into account the sign of the identification function $I()$, several individuals may have the same raw fitness. This is often the case at the end of the evolution, which causes a loss of diversity.

3.1.3 Clustering

The idea is to group individuals having the same *rawfitness* into clusters. Then, inside each cluster, individuals are sorted according to their number of nodes as described in figure 3. The first and best one is the one with the smallest number of nodes.

```

Input: population of size  $N$ 
Output: population of size lower or equal to  $N$ 
foreach cluster of the population do
  if size of the cluster > to_keep then
    | remove the last to_remove
    | individuals from the cluster
  else
    | keep all individuals from the
    | cluster
  end
end
Algorithm 1: Elimination
  
```

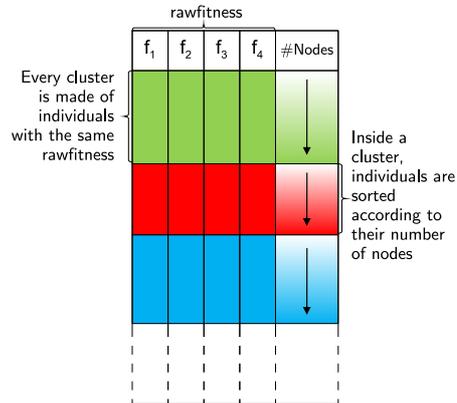


Fig. 3 Population clustering

3.1.4 Elimination rules

Useless individuals elimination allows to decrease the population size: An individual is considered as useless if it belongs to a big cluster and has a large number of nodes. The elimination rule depends on two parameters (*to_keep* and *to_remove*), in order to tune the decreasing speed of the population while keeping enough diversity. The elimination procedure is called at the end of each generation, the detailed procedure is given in algorithm 1: if a cluster has less than *to_keep* individuals, they are all kept, and if it has more, only the last *to_remove*, having the largest number of nodes, are removed. Typical values of these parameters are *to_keep* = 7 and *to_remove* = 1.

3.1.5 Fair play comparison

In order to fairly compare different schemes, results will be indexed with the number of new individuals evaluations instead of the number of generations. As a consequence, for the same cost (i.e the same total number of evaluations) a decreasing size population scheme “uses” more generations.

3.2 *Partial restart scheme: deflating and inflating the population*

3.2.1 Stagnation

As we are dealing with a Parisian approach, it is necessary to distinguish local and global levels:

- the adjusted fitness is used as a basis for selection, crossover and mutation operators, associated to a first elitism mechanism which keeps in the population the four best individuals *of the current generation* (one per phase) based on the non adjusted fitness.
- at the end of each generation, the global fitness is computed and reinjected in the population as a bonus, combined with a second elitism mechanism, which keeps the four individuals *of the generation that yielded the best global fitness*.

Despite of local elitism and bonus mechanisms, the global fitness is not a monotonically increasing function. In particular, it often happens that a generation notably improves the global fitness, while the generations that follow are not able to keep it.

3.2.2 Partial restart

In order to avoid stagnation due to over-specialisation of the best individuals, we propose to periodically add “fresh blood” to the population (i.e. new random individuals) if a stagnation criterion is fulfilled. The corresponding algorithm uses one

parameter denoted to_insert , typically set to a lower value than to_keep , see algorithm 2.

Input: population of size N
Output: population of size between N and N_{max}
 creation of a fresh population of $N_{max} - N$ individuals randomly created **foreach** individual of the fresh population **do**
 if size of cluster in which the individual fits $\leq to_insert$ **then**
 insert the individual into the corresponding cluster of the old population
 end
end

Algorithm 2: Partial restart

In this way, if a cluster of the old population is empty or has not enough elements according to a stricter rule than during the elimination process, it gets new elements. Moreover, the size of the subpopulation to be included being $N_{max} - N$, the final population is insured to be between N and N_{max} .

3.2.3 Criterion of stagnation

If the last improvement of the global fitness is older (in terms of generations) than $stagnation_threshold$, then the partial restart is triggered.

3.2.4 Deflation-inflation scheme

It is made of the following steps (see figure 4):

- **mutations and crossover** yield a temporary population $tmppop$
- **local fitness** is computed on the temporary population: $localfitness(tmppop)$
- **adjusted fitness** is computed via sharing: $sharing(pop + tmppop)$
- **selection** of the N best individuals: $pop = survival(pop + tmppop)$
- **elimination** of the useless individuals with algorithm 1: $pop = elimination(pop)$
- **global fitness** computation of the global fitness of the population: $globalfitness(pop)$
- **partial restart** if a stagnation criterion is met, using algorithm 2: $pop = restart(pop)$

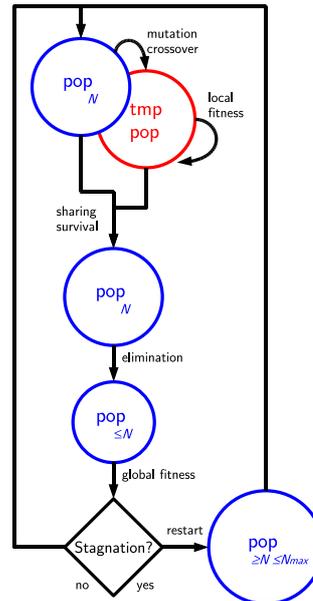


Fig. 4 Deflation-inflation scheme

4 Analysis

4.1 Experimental protocol

Available Camembert ripening data have been collected from 16 experiments during 40 days each, yielding 575 valid measurements.⁴ The derivatives of pH , la , Km and Ba have been averaged and interpolated (spline interpolation) for some missing days. Logarithms of these quantities are considered.

A statistical comparison between the three schemes is based on 100 runs. For each run, we share the 16 experiments into a learning set, made of 10 to 13 randomly chosen experiments, and a test set, made of the rest of the experiments. The three strategies are tested on the same sets during 50000 evaluations.

The parameters are detailed in table 2. The code has been developed in Matlab, using the GPLAB toolbox [12].

	Fixed size	Deflating-only	Deflating-inflating
Population size	1000	1000, then decreasing	1000, then decreasing and increasing
Clustering parameters	none	$to_keep = 7$ $to_remove = 1$	$to_keep = 7$ $to_remove = 1$ $to_insert = 3$
Number of evaluations	50000		
Sharing	$\sigma_{share} = 1$ on the first third of evaluations, then linear decrease from 1 to 0.1 $\alpha_{share} = 1$ (constant)		

Table 2 Parameters of the three strategies.

4.2 Results

Medians, means and standard deviations have been computed for the percentage of correct classifications on the test and learning sets (see figure 5). Number of evaluations and number of generations to reach the best individual, as well as total number of generations for 50000 evaluations are presented in figure 5 and table 3.

Using the fixed sized population as a reference for comparisons, one observes on table 3 that the deflating-inflating scheme allows to gain almost +2% on the test set, whereas the deflating-only scheme reaches almost the same score. The same conclusions can be drawn on the learning set. More precisely, on figure 5 it is to be noticed again that the classification on test set is better on average with the deflating-inflating scheme, but also that it has a narrowed range of values, i.e it fails less often.

⁴ The data samples are relatively balanced except for phase 3, which has a longer duration, thus a larger number of samples: We got 57 representatives of phase 1, 78 of phase 2, 247 of phase 3 and 93 of phase 4.

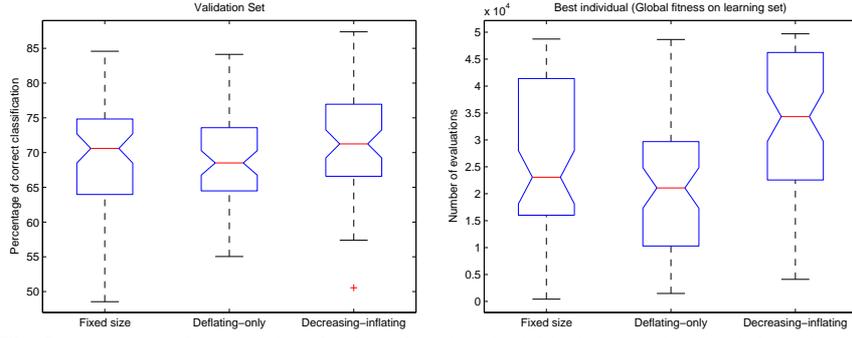


Fig. 5 Percentage of correct classification of the best individual on learning set (left) and number of evaluations needed to reach it (right), statistics are made on 100 runs.

	Fixed size			Deflating-only			Deflating-inflating		
	median	mean	std	median	mean	std	median	mean	std
Correct classification on the test set	70.59%	68.93%	8.48	68.51%	68.69%	7.32	71.24%	70.96%	7.95
Correct classification on the learning set	79.49%	79.39%	2.75	79.17%	78.76%	3.26	80.33%	80.09%	3.27
Number of evaluations (best individual)	23065	25866	14612	21073	20827	12727	34324	33130	13637
Number of generations (best individual)	39	41.10	22.93	46	68.29	94.74	70	70.06	32.72
Number of generations (total)	74	75.26	7.77	269	356.32	240.71	98	100.87	12.52

Table 3 Experimental results of the three strategies

As far as the number of evaluations is considered, one notices on table 3 and figure 5 that decreasing the size of the population and then increasing it enables to reduce the stagnation effect (the best individual is reached far later). This stagnation effect is more visible with the deflating-only scheme, due to the fact that decreasing the size of the population also decreases its diversity.

4.2.1 Analysis of variance

A one-way ANOVA has been used for comparing the means of the various test-samples⁵. It returns the p-value for the null hypothesis, that is “the two sets are samples of the same mean.” We compare strategies two by two, first fixed versus deflating-only, then fixed versus deflating-inflating, and finally deflating-only versus deflating-inflating. Results are given in table 4

	Fixed size VS Deflating-only	Fixed size VS Deflating-inflating	Deflating-only VS Deflating-inflating
Correct classification on the test set	0.8602	0.1627	0.0930
Correct classification on the learning set	0.2331	0.1921	0.0219

Table 4 P-values

⁵ This test supposes that the distributions of the samples are Gaussian, which is obviously not the case here. In the absence of additional hypotheses, the p-value however provides a quite good measurement of the similarities of samples distributions.

A large p-value (close to 1) corresponds to a high probability of having two samples of the same mean. This is the case for the classification on the test set for the fixed size and deflating-only schemes. While deflating-only and deflating-inflating have much lower p-values, meaning that there is a significant statistical difference.

5 Conclusions and future work

This work is a first attempt to manage varying population sizes within a Parisian GP scheme. The results presented above show the effectiveness of the population deflation-inflation scheme in terms of computational gain and quality of results on a real problem. The deflating scheme allows to obtain the same result as the fixed-size population strategy, but using less fitness evaluations. The deflating-inflating strategy improves the quality of results for the same number of fitness evaluations as the fixed-size strategy.

In general the development of a monopopulation cooperative-coevolution GP scheme is very attractive as it allows to evolve simpler structures during less generations, and yield results that are usually easier to interpret. However, as one “Parisian” generation necessitates more complex operations, one must carefully consider the global gain of such a procedure (in terms of fitness evaluation or even global computation time). The implementation of a population deflating-inflating scheme is another way to spare computational power, as it allows to avoid redundancy while regularly renewing population diversity.

More generally, the deflation-inflation scheme has two major characteristics: a clusterisation-based redundancy pruning and a selective inflation, which tries to maintain limited-size clusters with low complexity individuals. These two concurrent mechanisms tends to better maintain low complexity individuals as well as genetic diversity. These characteristics may actually be transposed to classical GP or EAs, in particular to limit GP-bloat effects.

Further work on this topic will also be devoted to the development of the Parisian deflation-inflation scheme to more complex problems related to agrifood process modeling within the INCALIN project.

References

1. O. Barrière, E. Lutton, C. Baudrit, M. Sicard, B. Pinaud, N. Perrot. Modeling human expertise on a cheese ripening industrial process using GP. 10th International Conference on Parallel Problem Solving From Nature, PPSN 2008, Dortmund, Germany, September 13-17, 2008.
2. D. Barile, J.D. Coisson, M. Arlorio and M. Rinaldi, Identification of production area of Ossolano Italian cheese with chemometric complex approach, *Food Control*, Volume 17, Issue 3, March 2006, Pages 197-206.
3. J. Bongard, H. Lipson, Active Coevolutionary Learning of Deterministic Finite Automata, *Journal of Machine Learning Research* 6, pp 1651-1678, 2005.

4. A. Bucci and J. B. Pollack, On identifying global optima in cooperative coevolution, GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary, Washington DC, USA, 2005.
5. Baudrit, C., Wuillemin, P-H., Sicard, M., Perrot, N., A Dynamic Bayesian Network to represent a ripening process of a soft mould cheese. *submitted*.
6. Pierre Collet, Evelyne Lutton, Frederic Raynal and Marc Schoenauer Polar IFS + Parisian Genetic Programming = Efficient IFS Inverse Problem Solving, *In Genetic Programming and Evolvable Machines Journal*, Volume 1, Issue 4, pp. 339-361, October, 2000.
7. L. Davis, Adapting Operators Probabilities in Genetic Algorithms. 3rd ICGA conference, 1989, Morgan-Kaufmann pp 61-69.
8. Kalyanmoy Deb and David E. Goldberg. An investigation of niche and species formation in genetic function optimization. In *Proceedings of the third Conference on Genetic Algorithms*, pages 42–50, 1989.
9. E. D. de Jong, K. O. Stanley and R. P. Wiegand, Introductory tutorial on coevolution, ECCO '07: Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation, London, United Kingdom, 2007.
10. A. E. Eiben, E. Marchiori, V. A. Valkò, Evolutionary algorithms with on-the-fly population size adjustment, Parallel Problem Solving from Nature PPSN VIII, LNCS 3242, 2004
11. David I. Ellis, David Broadhurst and Royston Goodacre, Rapid and quantitative detection of the microbial spoilage of beef by Fourier transform infrared spectroscopy and machine learning, *Analytica Chimica Acta*, Volume 514, Issue 2, 1 July 2004, Pages 193-201
12. Sara Silva, GPLAB A Genetic Programming Toolbox for MATLAB, <http://gplab.sourceforge.net/>
13. Ioannou, I. and Mauris, G. and Trystram, G. and Perrot, N. Back-propagation of imprecision in a cheese ripening fuzzy model based on human sensory evaluations. *Fuzzy Sets And Systems*, vol. 157, 1179–1187, 2006.
14. Jimenez-Marquez, S. A. and Thibault, J. and Lacroix, C. Prediction of moisture in cheese of commercial production using neural networks. *Int. Dairy J.*, vol. 15, 1156–1174, 2005.
15. F. G. Lobo, A review of adaptive population sizing schemes in genetic algorithms, In Proceedings of the 2005 Workshop on Parameter Setting in Genetic and Evolutionary Algorithms (PSGEA 2005), part of GECCO 2005.
16. Leclercq-Perlat, M.N., Picque, D., Riahi, H., and Corrieu, G. Microbiological and Biochemical Aspects of Camembert-type Cheeses Depend on Atmospheric Composition in the Ripening Chamber. *J. Dairy Sci.*, (89), pp. 3260-3273, 2006..
17. Leclercq-Perlat, M.N., Buono, F., Lambert, D., Latrille, E., Spinnler, H.E., and Corrieu, G. Controlled production of Camembert-type cheeses. Part I: Microbiological and physicochemical evolutions. *J. Dairy Res.*, (71), pp. 346-354, 2004.
18. Gabriela Ochoa, Evelyne Lutton and Edmund Burke, Cooperative Royal Road Functions, *In Evolution Artificielle*, Tours, France, October 29-31, 2007.
19. L. Panait, S. Luke and J. F. Harrison, Archive-based cooperative coevolutionary algorithms, GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation, Seattle, Washington, USA, 2006.
20. Bruno Pinaud, Cédric Baudrit, Mariette Sicard, Pierre-Henri Wuillemin, Nathalie Perrot, Validation et enrichissement interactifs d'un apprentissage automatique des paramètres d'un réseau bayésien dynamique appliqué aux procédés alimentaires, *Journées Francophone sur les Réseaux Bayésiens*, Lyon, France (2008).
21. E. Popovici and K. De Jong, The Effects of Interaction Frequency on the Optimization Performance of Cooperative Coevolution. In Proceedings of Genetic and Evolutionary Computation Conference, GECCO 2006.
22. Riahi, M. H. and Trelea, I. C. and Leclercq-Perlat, M. N. and Picque, D. and Corrieu, G. Model for changes in weight and dry matter during the ripening of a smear soft cheese under controlled temperature and relative humidity. *International Dairy Journal*, 17, 946–953, 2007.

23. D. Schlierkamp-voosen, and Heinz Mühlenbein, Adaptation of population sizes by competing subpopulation, In Proceedings of the 1996 IEEE Conference on Evolutionary Computation, Piscataway.
24. C.D. Tarantilis and C.T. Kiranoudis, Operational research and food logistics, *Journal of Food Engineering*, Volume 70, Issue 3, October 2005, Pages 253-255
25. R. P. Wiegand and M. A. Potter, Robustness in cooperative coevolution, GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation, Seattle, Washington, USA, 2006.