

---

# Emergence of Profitable Search Strategies Based on a Simple Inheritance Mechanism

---

**Natalio Krasnogor**

Intelligent Computer System Centre  
University of the West of England  
Bristol, United Kingdom  
Natalio2.Krasnogor@uwe.ac.uk  
<http://www.csm.uwe.ac.uk/~n2krasno>

**Jim Smith**

Intelligent Computer System Centre  
University of the West of England  
Bristol, United Kingdom  
James.Smith@uwe.ac.uk  
+44 117 3443357

## Abstract

In this paper we show how a simple inheritance mechanism is capable of learning the best local search to use at different stages of the search. In our work an individual is composed by its genetic material and its memetic material. The memetic material specifies the strategy the individual will use to do local search in the vicinity of the solution encoded in its genetic part. A simple vertical inheritance mechanism is enough to provide a robust adaptation of behavior. This result spans from a simple *OneMax* problem, to *NK-landscapes* and the *TSP*.

## 1 INTRODUCTION

In this paper we introduce a Memetic Algorithm (MA) in which the local search (a meme) employed by each individual is learnt during evolution. An individual is composed of its genetic material and its memetic material. The memetic material specifies the strategy the individual will use to do local search in the vicinity of the solution encoded in its genetic part. A simple vertical inheritance mechanism, as used in self-adaptive genetic algorithms and evolutionary strategies, is enough to provide a robust adaptation of behavior. We begin by illustrating the viability of the adaptive mechanism with two experiments where the GA adapts to use suitable mutation probabilities. With our method any meme, that is any mutation rate, is accessible with equal probability from any other one. This can not be achieved by a binary or gray encoding using multiple bits[1][17] nor with a real value encoding attached to the normal genes [8]. Also, by using this mechanism, the control of which memes to use is a distributed one. Furthermore, memes themselves can be modified by an

adequate mechanism. For a detailed review of operator adaptation refer to [18]. This is then expanded to a MA where the memes represent local search algorithms. In the memetic algorithms literature authors have spent a considerable amount of research assessing, e.g., how deep the local search should be and how often[7]. Land[13] used the concept of “sniffs” to try to gauge which individuals should go through a local search phase and with how much intensity. In [6] the authors developed a systemic model of Global-Local search hybrids that shed some light on the optimization of those algorithms. Carrizo et.al. in [5] employed several local searchers within the same MA to solve quadratic assignment problems. Moreover, to the best of our knowledge, just a few papers[19][10] have appeared where the choice of **which** local search to apply was left to the evolutionary process itself. It is in this spirit that this work is done.

## 2 THE MEMETIC ALGORITHM AND THE SIMPLE INHERITANCE MECHANISM

In this section we will describe the underlying GA architecture used in our experiments. An individual is composed of genetic material plus a meme allele. The genetic part was the representation of the potential solution. There were  $M$  memes available to be expressed by an individual, that is to say we treat our memes as categorical rather than ordinal entities. For the *OneMax* and *NK-Landscapes* Problems memes represented mutation strategies. In this case they were not associated with any local search process so we can regard our memetic algorithms as an adaptive GA. In the case of the *TSP*, memes were chosen from a range of local search strategies, embodying a fully fledged MA. The mutation process of an individual involves mutating its meme and its chromosome. The meme is mutated accordingly to a small innovation\_rate  $IR$  by ran-

domly choosing a meme number from the distribution  $U(1, M)$ . The  $IR$  takes a value in the range  $[0, 1]$ . A value of 0 means that there is no innovation and hence if a meme allele is lost it will not be re-introduced in the population. A value of 1 specifies an extremely explorative meme policy where all the different strategies implied by the available  $M$  memes will be equally used and no emergent properties are expected to arise. After that, the mutation strategy given by the meme is expressed. This mutation strategy specifies the kind of genetic mutation (One Point mutation or Bit Wise mutation) and the probability of applying it to the chromosome. The `innovation_rate` guarantees a minimum level of exploration of the memetic space. For an `innovation_rate` of  $IR$ , a population size of  $\mu$  and a uniform distribution of meme mutations  $U(1, M)$ , even the worst meme can be reintroduced to the population with a frequency of  $P_r = \frac{IR \times \mu}{M}$  per generation. Crossover is based on the following pseudocode:

```

Individual_Level_Crossover(parent1, parent2)
BEGIN
  IF(both parents carry the same meme)
    Cross parents genetic material.
    Inherit common meme to offspring.
  ELSE-IF (parent1.fitness()==parent2.fitness())
    /* the two parents have different memes          */
    /* but their fitness are comparable hence        */
    /* a random choice is made                       */
    Cross parents genetic material.
    Choose a meme randomly from any of the two parents.
    Inherit selected meme to offspring.
  ELSE
    /* parents don't share memes nor fitness values */
    /* hence the fittest individual                 */
    /* imposes its meme preference                  */
    Cross parents genetic material.
    Choose meme from fittest parent.
    Inherit the chosen meme to offspring.
END

```

The first phase involves the standard chromosome crossover, while the second phase performs the vertical propagation of the memes in the following way. If two individuals share the same meme then this meme will be inherited to the offspring. If the memes they carry are different, then the meme of the fittest parent is propagated. Finally, if memes are different but the fitnesses are equal, then a random choice between both memes will be done and the selected one will appear in the offspring. The memetic phase of the crossover is kept identical in the three problems. The first phase of chromosome crossover are different; for the *OneMax* uniform crossover with probability 0.7 was used, for the *TSP DPX* with probability 0.6 was used. In the case of *NK-Landscapes* no chromosome crossover was employed. As we said before an individual consists of its chromosome and its meme. This meme specifies a strategy that is composed of both an operator and its probability of being applied. These composed memes are called ‘memeplexes’ (see for example [3]). In the

first two experiments presented the meme encoded a fixed mutation operator with variable probabilities for the binary problems, while for the *TSP* the meme represented variable local searchers with a probability of being applied fixed at 1.0 .

### 3 EVOLUTIONARY ACTIVITY WAVE AND MEME CONCENTRATION GRAPHS

In order to examine the evolutionary and adaptive properties of memes in our system we will use the approach of Bedau et.al. [2]. We are interesting in observing the adaptive significance of the search strategies coded by the memes. We define the *concentration*  $c$  of meme  $i$  at time  $t$  as the number of individuals in the population that carry this meme. We denote this value by  $c_i(t)$ . It is hypothesized that since memes are carried alongside genes, those strategies that confer a selective advantage to the genes (i.e. they represent an efficient local search) will proliferate. Moreover, this proliferation is going to be reflected as an increase on those meme’s concentration. The meme concentration  $c_i(t)$  is a crude measurement of a meme success because it doesn’t give information about its continual usage since it first appeared in the population. Some memes might have a low concentration at a given time but disappear or take over the population in the next few generations. To account for such phenomena the *evolutionary activity (E.A.)*  $a$  of meme  $i$  at time  $t$  is defined by:

$$a_i(t) = \begin{cases} \int_0^t c_i(t) dt & \text{if } c_i(t) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

When the evolutionary activity of a meme is plotted, the slope of the curve represents the concentration  $c_i(t)$  of meme  $i$ . In that way, an increase in the slope indicates an increasing use of a given meme (it is spreading fast in the population). An almost flat curve points out a meme that is having less survival value than its competitors. In this kind of graph is usual to distinguish a wave of activity when a meme that has been successful for many generations disappears.

### 4 THE *OneMax* AND *NK-Landscapes*

In this section we describe several experiments performed to understand the behavior and the feasibility of adapting memes in a population of evolving individuals. We will describe and analyze the results of several experiments on two different, yet related, prob-

lems: *OneMax Problem* and *NK-Landscapes*[9]. The *OneMax* problem consists of achieving an all ones bit string of length  $n$  starting from a randomly initialized population. The *NK-Landscapes* are binary problems of length  $n$  where genes participate in epistatic interactions. The number of genes with which any other gene interacts depends on  $K$ .

Furthermore, in the case of *OneMax*, in generation 370 (out of 1000) the problem was changed from maximizing the number of ‘ones’ to that of maximizing the number of ‘zeroes’. This change in the fitness function provides a dynamic environment where individuals (genes and memes) were tested against very different situations. At the beginning of the run the population was randomly initialized, with both genes and memes set randomly. After the environment transition the evolving population was faced with a new problem (that of *ZeroMax* instead of ones). In practice this was equivalent to restarting the experiment but with a non-random population. In this way we were able to study the behavior of our approach under three different regimes: A random starting one and its adaptation towards an optima, a transient state with a biased (converged) initial population, and a final converged state. Thirty runs were made with a generational GA with no elitism. Deterministic binary tournament was used to select parents. The population size was 50. Uniform crossover was used with a probability of 0.7. The 11 memplexes specified a one point mutation together with its ‘per individual’ probability of being used. The mutation probabilities were in the range [0.0, 1.0]. For this experiment  $IR = 0.1$ . In figure 1 we plot the average of the mutation probabilities in the memes that exist at time  $t$  in the population. Also the average fitness achieved by the population is shown.

The dominating meme corresponds to a 0 mutation probability. This meme is successful because it preserves whatever the evolutionary system achieved. At the fitness transition it is wiped out and memes that represent high mutation rates take over the population (memes with mutation probabilities of 0.6, 0.8 and 0.9). When the new problem, *ZeroMax*, is 50% solved (i.e. half the allele values are optimal) those memes rapidly disappear and again strategies that represent low mutation rates dominate the population. From the graph in figure 1 we can see that at the beginning of the run, when the population is randomly initialized and the goal is to maximize the number of ones, the average mutation probability expressed by the memes is around 0.5. At the fitness transition and because the population is biased towards all ones, the system tunes to a much higher average mutation: 0.7. When the system starts to maximize the number of ones on aver-

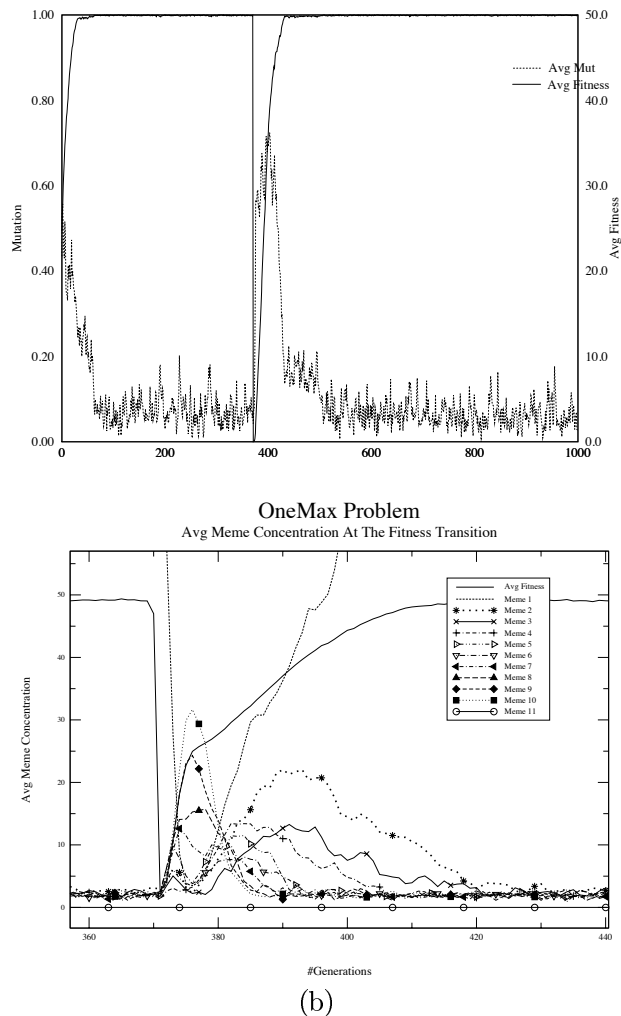


Figure 1: In (a) Average mutation rate expressed by the individuals in the population at time  $t$ . The mutation rate is defined by the meme each individual keeps, higher meme number means higher mutation rate. Average fitness is in solid line. Fitness transition is at generation 370. (b) Average meme concentration in the system at the fitness transition. Memes with higher numbers dominate (high mutation rates)

age half the bits will be properly set. After achieving an average fitness of at least 50%, mutations become deleterious and selection works against mutation. On the other hand, when the system switches to maximizing the number of zeroes, the majority of allele values are suboptimal, so with high probability mutating a bit will give a selective advantage. Thus, those memes which correspond to a higher probability of achieving this advantage will flourish until the mean density of zeroes is greater than 50%. Furthermore the expansion of the high mutation rates is longer than the one at the beginning of the run. In figure 1(b) the average meme concentration can be seen. The patterns of concentration at the start of the search and during the fitness transitions differ from each other (not shown here). During the fitness transition memes with even higher mutation rates are favored and it takes longer for their concentrations to decrease. This experiment demonstrates the ability of a system with simple meme encoding to adapt, even though (unlike other approaches) memes are not treated as continuous or ordinal entities. A theoretical model of this method can be seen in [16].

## 5 ADAPTATION AND PHASE TRANSITIONS IN PARAMETER SPACE

In the previous sections we described the main architecture of our adaptive GA and we showed that it was capable of tracking changes in the environment by appropriately tuning the mutation rates of the evolutionary search. We also conducted a series of experiments with *NK-Landscapes* to assess if our system was able to adapt the mutation rates in more complex settings. In [14] the authors explore a phase change in search when a parameter  $\tau$  reaches a certain critical value on some *NK-Landscape* problems. In their experiments the authors focused on Simulated Annealing (SA) as a local search algorithm, although a very special SA: the temperature was kept equal to zero at all times. The underlying operator, a bit-flip, was parameterized with  $\tau$ , a per bit mutation rate. In their paper the authors show experimentally that the quality of the search follows an *s*-shape curve when plotted against  $\tau$  making evident a change in phase. We wanted to explore whether the same kind of phenomenon arises in a GA and, if indeed this was the case, if our adaptive mechanism was able to select mutation rates comparable to those before the demeliorating transition.

As a first step we ran extensive simulations of the GA behavior with different bit rate mutations covering a wide range of values. The same GA as before

was used but with a zero probability of crossover and 100 generations. A set of experiments was done with  $N = 40$  and  $K \in [0, 15]$ . For each  $K$  three landscapes were created and 10 runs made on each landscape. This was repeated for 29 mutation rates in  $\{0.0005, 0.0010, \dots, 0.0045\} \cup \{0.005, 0.010, \dots, 0.10\}$ . In the upper part of figure 2 we can see the results obtained<sup>1</sup>. The GA is sensitive to the per bit mutation probability, there is a change in behavior at  $\tau = 0.01$ . When  $\tau$  is further increased a swift loss of performance occurs. This critical value  $\tau_c$  is very close to the theoretically predicted error threshold[15] for finite asexual populations:

$$\tau_S^* = \frac{\ln \sigma}{\nu} - \frac{2 * \sqrt{\sigma - 1}}{\nu * \sqrt{S}} + \frac{2 * \ln \sigma * \sqrt{\sigma - 1}}{\nu^2 * \sqrt{S}} \quad (2)$$

where  $S$  is the population size,  $\nu$  is the genome length,  $\sigma$  is a “selective ratio” that gives a rough measure of fitness superiority of the master sequence. In our experiments  $S = 50, \nu = N = 40$  and  $\sigma = 2.0$ , the resulting  $\tau_S^* = 0.0103$ .

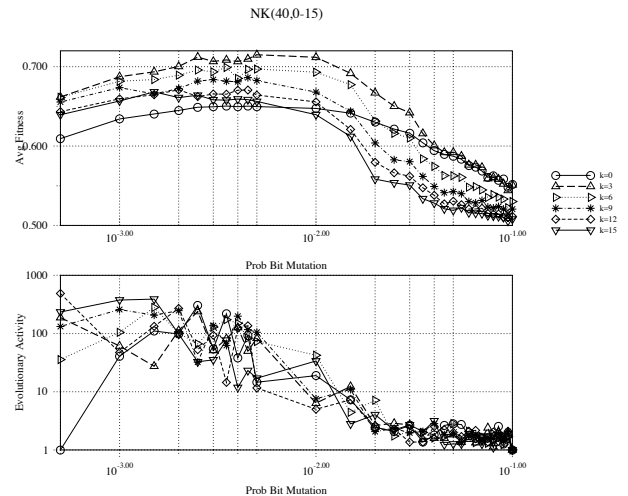


Figure 2: Up: Fitnesses achieved under different mutation rates for varying  $K$  averaged over 30 runs. Note the sharp decrease in fitness for mutations higher than 0.01. Down: Evolutionary activity of memes representing the same range of mutation probabilities as above. Note the fast decrease in activity for those memes above 0.01.

Once we knew that the phenomenon was present in GAs, we needed to check if the mechanism proposed here was capable of avoiding effective mutation rates equivalent to those greater than  $\tau^*$ . In our experiment we used 50 individuals per generation, the GA was a

<sup>1</sup>For clarity we show here just a few  $K$  that span the range studied

generational one and the memes encoded per bit mutation rates in the range described above and a zero probability of crossover. The problems used were as in the exhaustive experiments. In the lower part of graph 2 we see the evolutionary activity of memes, as defined by equation 1, for  $K$  in the same range as before. The graph shows the activity for generation 100. Memes were associated with the range of probabilities with which the exhaustive runs were performed. We can see that the adapting GA was able to distinguish between memes before and after the  $\tau^*$ . This is shown by the rapid decrease in evolutionary activity for those memes lying beyond 0.01. A second important conclusion that we can draw is that this simple adaptive mechanism is sensitive enough to be able to discriminate between a large set of alternatives (29 in this case) and it allows the emergence of effective mutation rates that avoid been trapped after  $\tau^*$ . In figure 3 we plot, for different  $K$ , the best fitness obtained from all of the exhaustive runs of the standard GA, the fitness of the adaptive GA and the fitness of the standard GA after the transition. As it can be seen in the graph the adapting GA, by differentially propagating memes that are before and after the transition, can sustain fitness values comparable to the optimal ones. As  $K$  increases, the gap between the adaptive and the optimal value decreases, while the gap with the values after the transition gets larger. Another observation is that the transition is not sharp for  $K \leq 4$ , confirming Macready's et.al. findings [14]. Important to note is the fact that the best fitnesses obtained for the standard GA (before and after the transition) were obtained with different mutations rates and involved 30x29 runs. The adaptive version achieves its results just with 30 runs.

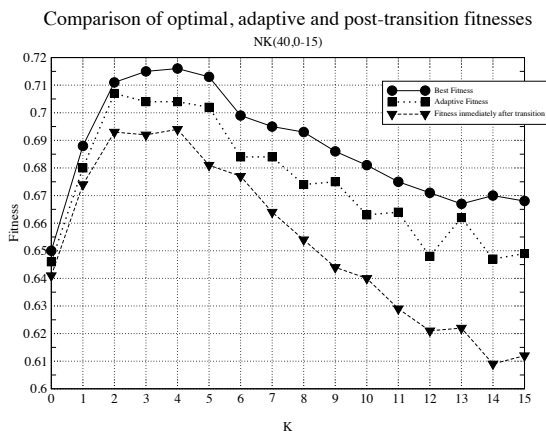


Figure 3: The best fitness for each  $K$  is compared to the fitness achieved by the adaptive GA and the fitness immediately after the transition at  $\tau^* = 0.0103$ . The values on this curves are averaged over 30 runs

## 6 ADAPTING THE BEST MEME FOR THE TSP

In the previous sections we showed that a simple vertical inheritance of memes was capable of performing an efficient adaptation of behavior for the dynamic *One-Max* and *NK-Landscapes*. In this part of the work we applied the same principle to learn which is the best meme to employ during different stages of the search for *TSP*. The *TSP* consists on finding the minimum length closed circuit among all the cities of a predefined set. The circuit should touch each city only once. We used 24 different memes, each meme defines the acceptance strategy, the underlying basic move and the number of iterations to use during the local search stage. There were two acceptance strategies, namely *first-improvement* and *best-improvement*. Three basic moves were considered *2-exchange*, *3-exchange* and *4-exchange*. The final property of a meme was the number of times the acceptance strategy was going to be iterated employing the basic move. We can represent a meme  $M$  by the three values that specifies its basic move ( $M$ ), its acceptance strategy ( $FB$ ) and its number of iterations ( $I$ ): *MeFBbIn*. The range of  $e$  was  $\{2, 3, 4\}$  implying a *2-exchange*, *3-exchange* or *4-exchange*. To specify a *first-improvement* acceptance strategy  $b$  was set to 1, and when  $b = 2$  then the meme used a *best-improvement* acceptance strategy. Finally,  $n$  gives the number of iterations drawn from the set  $\{1, 3, 6, 9\}$ . Because our goal in this paper is to see if this simple memetic system can learn the best meme to use and not to discover the best meme for a particular *TSP* instance, we assume that the execution cost of all of the 24 memes is equivalent. The reader should note that the memes with  $b = 2$  require greater computational cost than their counterparts  $b = 1$ . Furthermore, except for the *2-exchange* (for which the neighborhood explored by the acceptance strategy was complete), just a sample of the induced neighborhood was considered for the other moves. For all the experiments run the probability of mutation was 0.4, that of crossover 0.6 and the innovation rate was set to 0.125. The crossover used was DPX and the mutation operator the double-bridge move. The underlying GA was a generational GA with a (50,200) strategy with a tournament size of 4. The architecture of the MA was, according to [11], a  $D = 4$  MA, that is, local search was executed independently of mutation and crossover in a separate stage. The probability of local search (expressing the meme) was 1. The encoding used was a permutation encoding.

We first ran a set of experiments (one for each of the 24 memes), each consisting of 30 trials, where the whole

population used the same meme, that was fixed during the complete run. The goal of this experiment was to obtain a ranking of memes for the different instances. We then ran an adaptive MA where the meme alleles were evolved. The graph in 4(a) shows the evolution of fitness over time for different memes on the *lin318.tsp* instance from TSPLIB. The reader must keep in mind that the memes were executed within the underlying MA described above.

An ANOVA analysis of the average over 30 runs for the best tour in each experiment shows that the curves are (with 95% confidence level) different. The ANOVA, together with the post-hoc *t - test*, provide a sound ranking of the various memes. As can be seen in figure 4(a) the MultiMeme MA, that is, the memetic algorithm for which the adapting process was enabled, was able to closely follow the performance of the best meme. It achieves this by favorably selecting the memes that produce the best increment in fitness. This is shown by the evolutionary activity graph in figure 4(b). The same results, with statistical significance, were obtained for other instances of different size and nature: *eil76.tsp*, *lin105.tsp* and *mnpeano44.tsp*. We ran extensive experiments with a MultiMeme MA where the memes available were of the form *MeFB1In* with  $e \in \{2 - exchange, 3 - exchange, 4 - exchange\}$  and  $n \in \{1, 3, 6, 9\}$ . The algorithm was able to positively select the best meme and to match the performance of the best one (with statistical significance). As the size of the instances increased the memes were more easily differentiated, and MultiMeme was able to track the curve of the best meme. For the instances studied the evolutionary activity diagrams show<sup>2</sup> that while the evolutionary search is not yet stagnated and the search is progressing, just one or two evolutionary waves are conspicuous, while the other memes remain under spurious activity. The use of an  $IR > 0$  means that memes have non-zero background activity even if they are actually selected against (see section 2). When the search is converging towards local optima then several memes become neutral to each other and the evolutionary waves starts to develop. From our experiments it turns out that the best meme for all the instances was *M2FB1I9* and the second best *M2FB1I6*. This is not surprising since those memes perform 9 and 6 iterations respectively based on the complete neighborhood of a 2 - *exchange* while for the remaining moves the neighborhood was sampled. This fact led us to design an experiment where the memes involved where the same as before **except** that the first best

and the second best were not allowed to appear in the population. This MA will be called multiMeme-b. The results are shown in figure 5(a) for instance *lin318*.

After analyzing the results obtained, we observed that multiMeme-b was able to track and follow the curve of the best meme for the instance *mnpeano44.tsp*, however it fails to do so for *lin105.tsp* and *lin318.tsp*. In the later case the best meme, that is, the meme that at the end of the run produces the best fitness was *M2FB1I3*. The algorithm fails to select this one in favor of *M4FB1I9*, *M3FB1I9*, *M3FB1I6* and *M4FB1I6*. The reason for this behavior is simple to state: it pays for an individual to carry the meme that produces the maximum increase in fitness at any given point in time. Given that the individuals have no foresight of which is going to be the best fitness at the end of the run<sup>3</sup>, the meme that produces the behavior with the steepest increase in fitness (decrease in tour length) is favorably selected. However, as generations go by, the relative payoff of the different memes change. In the case of the *TSP* the reason for this dynamic payoff is rooted in the so called “Big Valley” structure. In Boese’s work [4] it is shown that the *TSP* shares with other commonly studied NP-Hard combinatorial optimization problems a globally convex structure of the set of local minima, where the local minima are points in the landscapes defined by different local search heuristics. The author shows that tours found by better heuristics are on average closer to each other in terms of distance<sup>4</sup> to the optimal solution, giving rise to the “Big Valley” metaphor. The gradient of improvement for the different memes changes during evolution while approaching a local optimum (eventually a global optimum). The adaptive MA, through its simple inheritance model is sensitive to this changes. Looking at the graph (b) in figure 5 we can see that the evolutionary wave of meme *M4FB1I9* is becoming almost flat. As explained in the previous sections this means that the selective advantage of carrying this meme is decreasing. Also, we can see that waves of evolutionary activity arise for memes *M3FB1I9*, *M3FB1I6*, *M2FB1I3*. Tracing back the origins of these waves it is possible to note that they match the time when the corresponding curves in graph (a) to the left surpass the curve of multiMeme-b. Three vertical bars are marked in the graph with  $x1, x2, x3$ . Moreover, the longer the simulation, the closer the gap between the curve of the best meme and the multiMeme-b approach. The same behavior was notice for the instance *lin105.tsp*. The

<sup>2</sup>Only the graph for *lin318.tsp* is shown due to space limitations

<sup>3</sup>The system is not teleological.

<sup>4</sup>Distance here is actually measure as the number of links that differentiate two tours

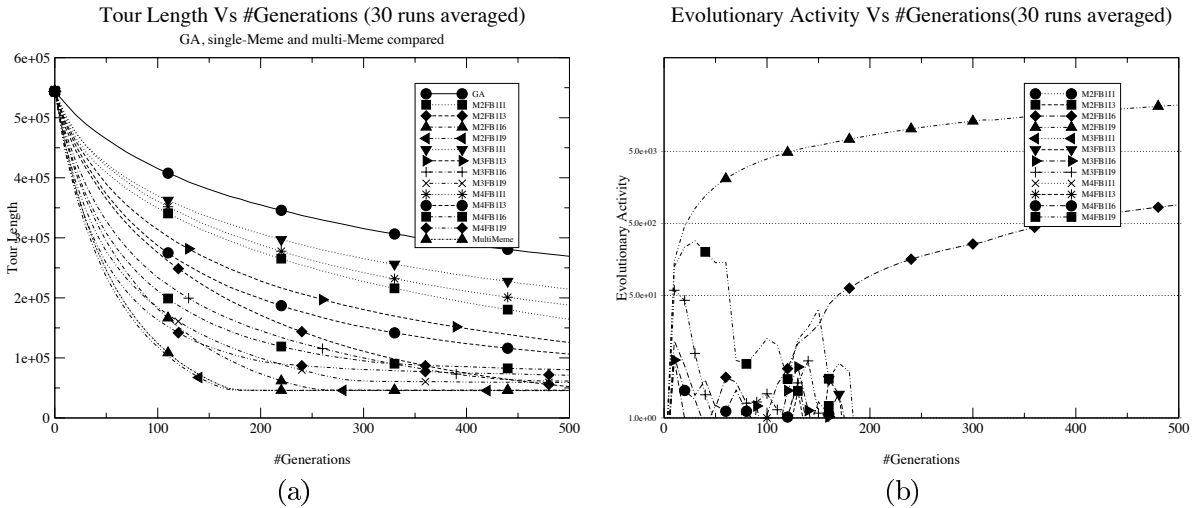


Figure 4: (a) Performance of different MAs when memes varies. Just the curves for the *first-improvement* strategy are shown. It is possible to see how the MultiMeme MA follows closely the performance of the best meme. In (b) the evolutionary activity of the MultiMeme MA is shown

reader should also note how the suppression of the best and second best memes alters the evolutionary activity diagram by comparing figure 4(b) and 5(b).

## 7 CONCLUSIONS

In this paper we showed how a simple vertical inheritance mechanism is enough to adapt the behavior of individuals in a memetic algorithm under different problems. Individuals have access to a set of memes that represent different search strategies. The evolutionary mechanism ensures that memes that are useful will be selected and spread in the population. Our approach differs from others (i.e. [19]) in that we are learning the association between an individual and a memplex and not a vector with the particular characteristics of a given meme in a set of memes. Hence, the dimensionality of the problem is much smaller. From an engineering point of view this is a sound approach because we can allow memes to change using any algorithm that we find suitable, i.e., we can run a GA to define the memes themselves. By isolating the structure of a meme from its phenotypic action in the genes we are facilitating the search in both genes and memes spaces. We tried our MA under three different scenarios. The dynamic *OneMax* problem showed that the adaptive MA was able to track changes in the environment, i.e. the fitness function, by triggering high mutation rates. We saw that for the *NK-Landscapes* the adaptive mechanism was robust enough to adapt to the edge of the transition after which mutation rates become pernicious. It was able to express an almost optimal mutation and to track closely the optimal fit-

ness achieved by exhaustive runs. In the case of the *TSP* we are able to conclude that the adaptive MA is capable of selecting the memes that provide the best performance at any given moment of time. As a by product of this paper we were able to show that the phenomena described by Macready et.al. for simulated annealing [14] is also present in GAs. The memes exploited in this paper were, accordingly to [11], static memes. We are currently running experiments with adaptive memes like those of [12]. Self adapting memes for the Protein Folding Problem will be studied soon.

## References

- [1] T. Back. Self adaptation in genetic algorithms. In F. Varela and P. Bourgine, editors, *Towards a Practice on Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 263–271. MIT Press, 1992.
- [2] M. Bedau and N.H.Packard. *Measurement of Evolutionary Activity, Teleology and Life*, volume 98-03-023. Addison-Wesley, 1992.
- [3] S. Blackmore. *The Meme Machine*. Oxford University Press, 1999.
- [4] K. Boese. *Models For Iterative Global Optimization*. Ph.D. Thesis, UCLA Computer Science Department, 1986.
- [5] J. Carrizo, F. Tinetti, and P. Moscato. A computational ecology for the quadratic assignment problem. In *Proceedings of the 21st Meeting*

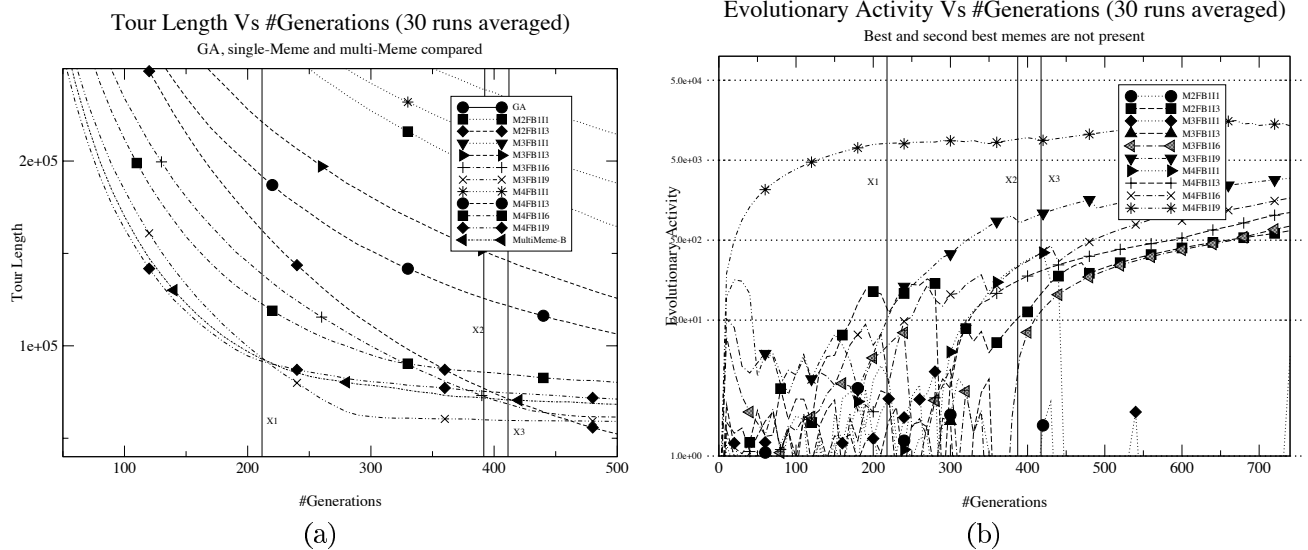


Figure 5: (a)Zoom-in in the performance graph. MultiMeme-b is a MA with all the memes available except the first best and second best. In (b) the evolutionary activity of the MultiMeme-B MA is shown

on Informatics and Operations Research, Buenos Aires, 1992. SADIO.

- [6] D. Goldberg and S. Voessner. Optimizing global-local search hybrids. In W. Banzhaf, J. Daida, A. Eiben, M. Garzon, V. Honavar, M. Jakaiela, and R. Smith, editors, *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*. Morgan Kaufmann, 1999.
- [7] W. E. Hart. Adaptive global optimization with local search. *Ph.D. Thesis, University of California, San Diego*, 1994.
- [8] R. Hinterding, Z. Michalewicz, and T. Peachey. Self-adaptive genetic algorithms for numeric functions. In *Proceedings of Parallel Problem Solving from Natur - PPSN IV*, 1996.
- [9] S. Kauffman. *The Origins of Order, Self Organization and Selection in Evolution*. Oxford University Press, 1993.
- [10] N. Krasnogor. Co-evolution of genes and memes in memetic algorithms. In A. Wu, editor, *Proceedings of the 1999 Genetic And Evolutionary Computation Conference Workshop Program*, 1999.
- [11] N. Krasnogor and J. Smith. Memetic algorithms: Syntactic model and taxonomy. submitted to The Journal of Heuristics. Available from the authors.
- [12] N. Krasnogor and J. Smith. A memetic algorithm with self-adaptive local search: Tsp as a case study. In *Proceedings of the 2000 Genetic and Evolutionary Computation Conference*. Morgan Kaufmann, 2000.
- [13] M. Land. Evolutionary algorithms with local search for combinatorial optimization. *Ph.D. Thesis, University of California, San Diego*, 1998.
- [14] W. Macready, A. Siapas, and S. Kauffman. Criticality and parallelism in combinatorial optimization. 1995. to appear in Science.
- [15] G. Ochoa and I. Harvey. Recombination and error thresholds in finite populations. *Foundations of Genetic Algorithms - 5*, 1999.
- [16] J. Smith. Modelling gas with self adaptive mutation rates. In *GECCO-2001: Proceedings of the Genetic and Evolutionary Computation Conference*. Morgan Kaufman, 2001.
- [17] J. Smith and T. Fogarty. Self adaptation of mutation rates in a steady state genetic algorithm. In *Proceedings of the Third IEEE International Conference on Evolutionary Computing*, pages 318–323. IEEE Press, 1996.
- [18] J. Smith and T. Fogarty. Operator and parameter adaptation in genetic algorithms. *Soft Computing*, pages 81–87, 1997.
- [19] H. Terashima-Marin, P. Ross, and M. Valenzuela-Rendon. Evolution of constraint satisfaction strategies in examination timetabling. In *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*, 1999.