
A Memetic Algorithm With Self-Adaptive Local Search: TSP as a case study

Natalio Krasnogor

Intelligent Computer System Centre
University of the West of England
Bristol, United Kingdom
natk@ics.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
jim@ics.uwe.ac.uk

+44 117 3443357

Abstract

In this paper we introduce a promising hybridization scheme for a Memetic Algorithm (MA). Our MA is composed of two optimization processes, a Genetic Algorithm and a Monte Carlo method (MC). In contrast with other GA-Monte Carlo hybridized memetic algorithms, in our work the MC stage serves two purposes:

- when the population is diverse it acts like a local search procedure and
- when the population converges its goal is to diversify the search.

To achieve this, the MC is self-adaptive based on observations from the underlying GA behavior; the GA controls the long-term optimization process.

We present preliminary, yet statistically significant, results on the application of this approach to the *TSP* problem. We also comment its successful application to a molecular conformational problem: *Protein Folding*.

Paper Category: Genetic Scheduling and TSP

1 Introduction

Memetic Algorithms, and Genetic Algorithms in general, have been applied in a number of different areas and problem domains. It is now well established that it is hard for a ‘pure’ Genetic Algorithm to ‘fine tune’ the search in complex spaces. Researchers and practitioners have shown that a combination of global and local search is almost always beneficial.

In Richard Dawkins’ book “The Selfish Gene” [7], the concept of a meme (a cultural gene) was introduced.

The memes reshape the search space where they interact with genes and can themselves adapt. Inspired by the ideas of learning in an evolutionary time scale and individual’s life-span time scale, Genetic Algorithms which used some kind of interaction with local searchers (adaptive or not) were named Memetic Algorithms (MAs). In different contexts and situations, Memetic Algorithms are also known as Hybrid GAs, Genetic Local Searchers, Baldwinian GAs, Lamarckian GAs, etc.

From an optimization point of view MAs have shown that they are orders of magnitude more accurate than traditional GAs for some problem domains. See for example reference [11] for a continuous domain research and [18] for combinatorial optimization studies. It is argued that the success of MAs is due to the trade-off between the exploration abilities of the underlying GA and the exploitation abilities of the local searchers used. The price to be paid is a greater number of fitness evaluations and often a swift loss of diversity within the population.

In this paper we introduce a promising hybridization scheme for a Memetic Algorithm (MA). Our MA is composed by two optimization processes, a Genetic Algorithm (GA) which controls the long-term behavior of the scheme, and a Monte Carlo like local search and diversification process. In our study the local search process pursues two goals. At the early stages of the global search it tries to focus, hence exploit, the vicinity of the point from where it starts the search. At later stages, when the population starts to converge or when the search is stagnated, it changes its role from exploitation to exploration, allowing the search to (eventually) move ‘uphill’ (if we assume the underlying GA is moving ‘downhill’ or minimizing) to explore different basins of attraction.

We applied the method to two NP-Hard problems, the *Traveling Salesman Problem (TSP)* and the *Protein*

2 TSP: Problem Definition, Previous Works and Methodological Approach

The *TSP* is one of the most studied combinatorial optimization problems. It is defined by *Minimum Traveling Salesman Problem*

Instance: Set C of m cities, distances $d(c_i, c_j) \in \mathbb{N}$ for each pair of cities $c_i, c_j \in C$.

Solution: A tour of C , i.e., a permutation $\pi : [1 \dots m] \mapsto [1 \dots m]$.

Measure: The length of the tour, i.e., $d(\{c_{\pi(m)}, c_{\pi(1)}\}) + \sum_{i=1}^{i=m-1} d(\{c_{\pi(i)}, c_{\pi(i+1)}\})$.

In [1] a short review on early MAs for the TSP was presented. Those works reported near optimal solutions for small instances of the problem. Although the results were not definitive they were very encouraging and many of the following applications of MAs to the *TSP* (and also to other NP-Optimization problems) were inspired by those early works.

In [12] a MA is used with several non-standard features. In [12] the local search used is based on the powerful Guided Local Search (GLS) meta-heuristic due to Voudoris and Tsang [27]. This algorithm was compared against Multi Start Local Search (MSLS), a GLS and a MA. The MA used as a local search engine the same basic move used by GLS but without the guiding strategy. In this paper experiments were run using instances taken from TSPLIB [25] and fractal instances [19]. In no case was the MSLS able to achieve an optimal tour, while the other three approaches were able to find optimal solutions. Out of 31 instances tested the MA solves to optimality 24, MSLS 0, MA with simple local search 10 and GLS 16. It is interesting to note that the mentioned paper did not intend to be a “better than” paper but a pedagogical paper where the MAs were exposed as a new meta-heuristic in optimization. For details the reader is referred to [12] and [22].

Merz and Freisleben in [9],[10] and [20] show many different combinations of local search and genetic search for the *TSP* (and the asymmetric version *ATSP*) while defining specific purpose crossover and mutation operators. In their approach the initial population was a set of local optima with respect to the Lin-Kernighan heuristic, which seeds the underlying GA with near optimal solutions at initialization. An important feature of their approach is that the selection strategy is not a $(\mu + \lambda)$ nor a (μ, λ) but a hybrid between the two.

As the authors remark, the *Large Step Markov Chains* and *iterated-Lin-Kernighan* techniques are special cases of their algorithm. In [20] their optimization scheme is changed to one which has a more traditional mutation and selection scheme. It is important to remark that Merz and Freisleben MAs are perhaps the most successful meta-heuristics for *TSP* and *ATSP* and a predecessor of the schemes described was the winning algorithm of the *First International Contest on Evolutionary Optimization*.

In [23] Nagata and Kobayashi described a powerful MA with an intelligent crossover in which the local searcher is embedded in the genetic operator. Watson et.al. in [28] described a detailed study of Nagata’s and Kobayashi works and relate it with the local searcher used by Merz et.al.

In what follows we will describe our MA for the TSP and our methodology. Our purpose here is to show the potential for both search and diversity in our approach. It is not the goal of this paper to develop a specialized TSP solver.

We have used very naive and generic genetic operators (i.e. crossover or mutation). The local search move that was employed used no knowledge of the instance being solved besides that provided by the fitness functions. In this way we can guarantee that any benefit found will be an intrinsic property of the approach and not the consequence of using powerful operators. In all of the previously cited MAs for the *TSP* intelligent operators were used in the form of specially designed mutations, crossovers and local searchers. Furthermore, the local searchers used in the cited papers employed knowledge about the instance (i.e. list of nearest neighbors, kd-trees, etc). Besides that, there is nothing in the literature like a “standard MA” for the TSP with which to make comparisons. For these reasons, “absolute values” comparisons with other approaches will be done elsewhere.

3 A Memetic Algorithm for the Traveling Salesman Problem

In this section we describe and study the new self-adaptive hybridization scheme. We will show how the nonlinear interaction between the underlying GA and the local search/diversification process (which is governed by observations from the GA’s population) gives rise to a better global search meta-heuristic.

3.1 Description of the Memetic Algorithm

The overall pseudo-code of the MA used is:

```

MA:
Begin
  Initialize population Parents;
  Repeat Until ( Finalization_criteria_met ) Do
    Local_Search(Parents,  $P_{ls}$ );
    mating_pool := Select_mating(Parents);
    Offsprings := Cross(mating_pool);
    Mutate(Offsprings);
    Parents := Select(Parents, Offsprings);
  Od
End.

```

In this basic scheme the *Select*(...) procedure is a $(\mu + \lambda)$ or a (μ, λ) selection strategy, representing two extremes of selection pressure, with the + -strategy having the highest pressure and the , -strategy the lowest. *Select_mating*(...) is a Tournament selection method. In the case of the + -strategy a given individual can be modified several times during its lifespan either by local search or by mutation because the strategy allows an individual to persist. The best individual is never modified by the local search method.

The local search/diversification process is:

```

Local_Search(Parents,  $P_{ls}$ ):
Begin
  /* Parents is a set of solutions to which local search */
  /* will be applied with prob  $P_{ls}$  */
  temperature =  $\frac{1}{\lceil \maxFitness - \minFitness \rceil}$ ;
  beginIterator(Parents);
  While ( ! endIterator(Parents) ) Do
    indip = getIterator(Parents);
    If ( $(p_{ls} \geq \text{random}(0, 1)) \wedge \text{indip} \neg \text{best solution}$ ) Then
      ApplyMove(indip);
    Fi
  nextIterator(Parents);
  Od
End.

```

The procedures *ApplyMove*(...) can be seen below. It must be noted that *Modify*(...) can be any local search move (i.e. a 2swap, city insertion, etc). The self adaptation of the local search to either an exploitation or exploration behavior is governed by the temperature parameter. In the case presented above the entire population shares the same temperature. This temperature determines the degree by which, uphill moves will be allowed. As the temperature is inversely proportional to the spread of fitnesses within the population, when the later converges the former rises. A consequence of this is that each individual in the pop-

ulation will be more “nervous” and will try to move away from its initial position, exploring the search space. Eventually, the fitnesses will spread lowering the population temperature. We prevent the modification by local-search of the best individual, hence the overall best fitness is always maintained.

```

ApplyMove(indip):
Begin
  /* This is a minimizing process */
  prevFitness = fitness(indip);
  Modify(indip);
  nFitness = fitness(indip);
  If (prevFitness > nFitness) Then
    Accept configuration;
  Fi
  Else
    deltaE = nFitness - prevFitness;
    threshold =  $e^{-k \cdot \frac{\text{deltaE}}{\text{temperature}}}$ ;
    If (random(0,1) < threshold) Then
      Accept configuration;
      /* even if worse than the previous one */
    Fi
  Else
    Reject changes;
  Esle
  Esle
End.

```

No parameter optimization was performed to set the underlying GA or the local search. Analyses like those surveyed in [4] for the *Number Partitioning Problem* using Simulated Annealing[21] can be used to tune the parameters of the local searcher to enhance its exploration and exploitation capabilities within the MA.

3.2 Instantiation of the MA to the TSP

We have applied *MA*(...) , *Local_Search*(...) and *ApplyMove*(...) (described in section 3.1) to the *TSP* with a modification in the definition of the *temperature* which was set to $\frac{1}{\lceil \maxFitness - \text{averageFitness} \rceil}$ to produce a smoother dynamic. The *Modify*(...) procedure used a *two_swap* (*TS*) move. The *two_swap* move selects a sub-tour and inverts its cities. This produces a 4 links change in a given tour. It is called *two_swap* because it changes two links of the original tour in favor of two new links. *Modify*(...) selects a random number (between 1 and 10% of the instance size) which specify how many chained applications of *two_swap*(...) will be applied to an individual. After applying the moves the new individual will be accepted following a Boltzmann distribution based on the current population temperature.

The details of the MA are as follow:

We used a population of 50 individuals. Crossover, mutation and local search were applied with probability 0.8, 0.05 and 1.0 respectively. These probabilities were kept fixed during the whole run. No exhaustive parameter optimization was done, but rather they were decided empirically based on a number of runs. The k value was 0.01 in the (μ, λ) strategy and 0.001 in the $(\mu + \lambda)$ strategy. Every individual performs a local search/diversification phase in each generation except the best individual. The mating selection strategy was Tournament selection of size 2, the GA was, in one case, a steady state GA with $(50 + 50)$ replacing strategy. In the other case a $(50, 50)$ selection strategy was used. The encoding used was an array of integers interpreted as follows: if position i has integer j then the link connecting city i to j , (i, j) , exists in the tour. The crossover used receives two parents and generates an offspring starting from a random city. It adds to the offspring the shortest edge, not yet in the tour, from either parents. If no edge from any parent is available then a random edge is added. The mutation was an application of the *two_swap* operator. The initialization of the population was random. The simulations were programmed in java using the Memetic Algorithm Framework (MAFRA¹)[17] and the simulations' code is available from the authors.

4 Experimental Method and Results

In this section we present the methods used and results obtained.

To test our approach we chose the instance *eil76.tsp* from TSPLIB[25]. This instance is one of no particular difficulty and it involves only 76 cities. We run 30 simulations under two different selection strategies, a $(50, 50)$ and a $(50 + 50)$ strategy. The former provides the weakest selection pressure and the later the strongest. We tried these two scenarios because we want to explore not only final tour length but also population diversity. We wanted to compare how well our self adaptive memetic algorithm performs under these two extremes. We test our algorithms against four other algorithms all of them sharing either of the selection strategies:

A **standard GA (GA)** with no local search of any kind, which constitutes the basis for constructing all the other algorithms tested (see section 3.2 for details). A **hill climber memetic algorithm (HC)** which used as local search the *two_swap(...)* move but only

¹MAFRA is a free package available to download from the author's URL.

Algorithms	GA	HC	BHC	LMA	MA
GA		-	+	+	+*
HC	+		+	+	+*
BHC	-	-		+	+*
LMA	-	-	-		+*
MA	-*	-*	-*	-*	

Table 1: Summary of statistical analysis for tour length under the $(50, 50)$ strategy: + denotes that the algorithm that names the row achieves a longer tour than the one that names the column, - denotes that the algorithm that names the row achieves a shorter tour than the one that names the column, - or + with * denotes statistical significant up to at least a p-value of 0.01

accepts improvements. A **boltzmann hill climber memetic algorithm (BHC)** which used the same decision procedures as the self adaptive memetic algorithm but with a fixed temperature. The temperature was set to be the average temperature employed by the self adaptive MA in one of its runs. A **linear annealing memetic algorithm (LMA)** which used the boltzmann criteria to accept/reject moves. In this case the temperature was set at the beginning of the run to a value that was linearly annealed during the run. And finally the **self adaptive memetic algorithm (MA)** as described in 3.1.

Each algorithm was run for 2000 generations, except the GA which was given 6000 generations. With this amount of generations the GA employed more fitness evaluations than all the other MAs. To compare the quality of our MA against the other four alternatives we look at two measures, the quality of the best individual at the end of the run and the diversity of the population at that time. The quality was equivalent to the tour length and the diversity the number of different fitnesses found in the population divided by the population size. We performed ANOVA and t-test analysis on the averages of these parameters over the 30 runs for the 5 algorithms. A total of 300 runs were analyzed ².

Tables 1,2,3 and 4 summarize the results obtained.

From table 1 we can see that the proposed MA achieves better final tour length than the standard GA, the GA with a Hill climber (HC) the GA with a Boltzmann Hill Climber(BHC) and the linear annealed MA. These anova results are of statistical significance with a p-value of at most 0.01. If we turn our attention to the diversity table in 2 we see that the self adaptive approach is capable of maintaining the diversity

²30 runs per each one of the 5 algorithms per each one of the two selection strategies.

Algorithms	GA	HC	BHC	LMA	MA
GA		+	-*	+	-*
HC	+		-*	-	-*
BHC	+	+		+	-*
LMA	-	-	-*		-*
MA	+	+	+	+	

Table 2: Summary of statistical analysis for population diversity under the (50,50) strategy: + denotes that the algorithm that names the row keeps a higher diversity at the end of the run than the one that names the column, - denotes that the algorithm that names the row maintains a lower diversity than the one that names the column, - or + with * denotes statistical significant up to a p-value of 0.05

Algorithms	GA	HC	BHC	LMA	MA
GA		-	+	+	+
HC	+		+	+	+
BHC	-*	-*		-*	-
LMA	-*	-*	+		+
MA	-*	-*	+	-*	

Table 3: Summary of statistical analysis for tour length under the (50+50) strategy: + denotes that the algorithm that names the row achieves a longer tour than the one that names the column, - denotes that the algorithm that names the row achieves a shorter tour than the one that names the column, - or + with * denotes statistical significant up to a p-value of 0.05

Algorithms	GA	HC	BHC	LMA	MA
GA		=	-*	=	-*
HC	=		-*	=	-*
BHC	+	+		+	+
LMA	=	=	-*		-*
MA	+	+	-*	+	

Table 4: Summary of statistical analysis for population diversity under the (50+50) strategy: + denotes that the algorithm that names the row keeps a higher diversity at the end of the run than the one that names the column, - denotes that the algorithm that names the row maintains a lower diversity than the one that names the column, - or + with * denotes statistical significant up to a p-value of 0.0007, = denotes equal diversity

of the population on higher values than the other four algorithms. The differences are again of statistical significance. This is saying that our method shows an important difference from the others when used under this particular selection strategy.

If we consider the +-strategy (tables 3 and 4) where the selection pressure is higher, the benefits of the self adaptive MA are evident. It obtains better final tour length values and sustains a higher diversity in the population than three of the four competitors. The boltzmann hill climber achieves a better final mean tour length than the adaptive MA, however the difference was not statistically significant. It is very interesting to note that the boltzmann hill climber was set up with a fixed temperature. This temperature was the mean temperature obtained from a run done by the adaptive MA. The boltzmann hill climber MA achieves the same quality results than the self adaptive one by maintaining a very high diversity in the population. Its diversity is statistically significant higher than all the other algorithms. The temperature with which the algorithms achieves its results is not known a priori and depends on the instance been solved, the operators, etc. The advantage of the self adapting version is that it will set itself in a regime where the temperature will oscillate around this mean value.

As mentioned in the introduction, the use of local search within a GA usually causes a premature convergence in the search space, hence maintaining a diverse population is crucial³. It can be seen from the lower/upper diagonal of tables 1,2 and 3,4 that in most cases, when an algorithm beats another in one table it beats (or is at least equivalent) the same one in the other table as well.

In a subsequent experiment we changed the encoding from the one described in 3.2 to a permutation encoding and used a PMX crossover keeping all the other parts of the 5 algorithms unmodified. Again, 30 runs of each algorithms under the two selection schemes were executed for 2000 (6000 in the GA case) generations. The results obtained were consistent with those shown above. The self adaptive MA is better in both final tour length and diversity of the final population with a statistical significant difference (not shown).

5 Application of the Approach to the Protein Folding Problem

We have also investigated the power of our approach on a different combinatorial optimization problem. It

³This is of particular importance on MA applied to dynamic optimization.

has been recognized in Boese’s Ph.D Thesis [5] 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 k-Opt(...) , Lin-Kernighan(...) , etc. This is known as the “big valley” solution space structure. The author shows that tours found by better heuristics are on average closer to each other in terms of distance to the optimal solution. Those are very convenient features for the searching heuristics. We then decided to run some experiments on a problem where apparently the optima are not distributed following the “big valley” picture, no equivalent to the powerful k-Opt(...) , Lin-Kernighan(...) , etc, heuristics are available. We choose the *Protein Folding* problem.

5.1 PFP: Problem Definition

Protein Folding is one the most exciting problems that computational biology faces today. In words of John Maynard Smith[26]:

“ Although we understand how genes specify the sequence of amino acids in a protein, there remains the problem of how the one-dimensional string of amino acids folds up to form a three-dimensional protein... it would be extremely useful to be able to deduce the three-dimensional form of a protein from the base sequence of the genes coding for it; but this is still beyond us.”

Because an “all-atom” simulations is extremely expensive researchers resort to simplified model of the *Protein Folding Problem*.

One of the most studied simple protein models is the hydrophobic-hydrophilic model (HP model) proposed by Dill [8]. The *PFP* in the *HP* model is defined by *Maximum Protein Folding*

Instance: A protein, i.e. a string over the alphabet $\{H, P\}$ ($s \in \{H, P\}^*$).

Solution: A self_avoiding embedding of s into a two(three) dimensional square(triangular) lattice (i.e. Z^2)

Measure: The number of H s that are topological neighbors in the embedding (neighbors in the lattice but not consecutive in s)

It has been shown to be NP-Hard for a number of formulations and models. See for example [3],[6],[2]. For space limitations we will not give a detailed discussion

of *PFP* here. The reader is referred to [14].

In our experiment we use the self adaptive MA as described in 3.1, but tailored with protein folding encoding and operators. The design criteria to tailor the MA to this problem were those analysed in [14]: Two-point mutation was used to change two consecutive values in a relative encoding of solutions with a probability 0.3 . One-Point, Two-Point and Uniform crossover operators were used with probability of 0.8. The probability of local search was set to 1. Furthermore, every pair of amino acids mapped to the same lattice position was penalized with a constant penalty, C , dependent on the length of the instance. In this MA a (500 + 500) selection strategy was used together with fitness proportional mating selection. The ApplyMove(...) applied a move (drawn from a set of operators) in each portion of a protein that was folded accordingly to certain pattern. If the pattern was found, then it was changed as the operator indicates:

- pivot moves
- unfold of substructure
- random macro-mutation of substructure
- reflection of substructure

these transformation were chosen based on [15],[14] and [24]. Each run of the MA consisted of 200 generations. As test cases we used twenty polymer sequences which have a relatively short length (less than 50 monomers). The instances[13] used are shown in table 5.

In all except instances 16,18,19 (for which 24 bond solutions where found) and for the 20th instance(for which solutions of 26 bond were found) the approach was able to reach optimal configurations. We run the same experiments with a GA without local search. Even though the GA was given the same amount of fitness evaluations it never reached optimal conformations for any of the instances studied. A Hill Climber MA using the same basic local search but without the self-adapting criteria suffered from an extremely fast lost of diversity which prevented the global search from achieving any optimal conformation.

6 Conclusions

In this work we have introduced what appears to be a promising hybridization scheme for a MA. Our approach blends a local searcher with a GA in such a

Seq. Nro	Sequence	Length	Opt.
1	HHPRHRPHRPH	12	11
2	HHPRHRPHRPHRPH	14	11
3	HHPRHRPHRPHRPHRPH	14	11
4	HHPRHRPHRPHRPHRPHRPH	16	11
5	HHPRHRPHRPHRPHRPHRPHRPH	16	11
6	HHPRHRPHRPHRPHRPHRPHRPHRPH	17	11
7	HHPRHRPHRPHRPHRPHRPHRPHRPHRPH	17	17
8	HHPRHRPHRPHRPHRPHRPHRPHRPHRPHRPH	20	17
9	HHPRHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPH	20	17
10	HHPRHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPH	21	17
11	HHPRHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPH	21	17
12	HHPRHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPH	21	17
13	HHPRHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPH	22	17
14	HHPRHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPH	23	25
15	HHPRHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPH	24	17
16	HHPRHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPH	24	25
17	HHPRHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPH	24	25
18	HHPRHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPHRPH	30	25
19	HHPRHRPH	30	25
20	HHPRHRPH	37	29

Table 5: Two dimensional triangular lattice HP instances, length and optimal energy value

way that they are coupled through an observable measurement of the population. Here it was the temperature of a Monte-Carlo like local search which was a function of the spread of fitnesses within the underlying GA's population. Other observables might be defined. With this approach we were able to obtain the benefits of the exploitation/exploration capabilities of MAs but avoiding their drawback of premature convergence and diversity crisis. For further analysis of this approach see [16]. Our approach was found to meet two goals: fine tuning the global search and diversifying the population. We tried the approach on two NP-Hard problems, *TSP* and *PFP*. For the *PFP* we were able to obtain optimal and near optimal molecular conformations using our MA. The same experiments switching off either the local search (standard GA) or the self-adapting feature (Hill Climber MA) never found a global optima on a set of preliminary experiments. Regarding the *TSP* experiments, we chose a small not extremely difficult instance of TSPLIB to conduct a series of experiments. Because our goal was to see the benefits of our approach, no intelligent operators (crossover, mutation, local search) were used, but rather simple ones. We were able to verify that the global search performed by our MA was following the intended behavior.

We can conclude that the approach is a promising avenue of research with the potential to obtain better results than a GA without needing any other mechanism to maintain diversity in the population.

7 Future Work

Much more experimentation and data analysis should be applied, not only with different size and complexity of instances of *TSP* and *PFP* but also with other problems. Special attention should be given to the be-

havior of our MA in the presence of instances that lie on the phase transition region of the instance space.

The application of the approach with specially designed operators and encodings will be explored.

A natural variation of the scheme presented above is one where every individual in the population has its own temperature and the local search/diversification process is applied according to it. Results will be published elsewhere.

References

- [1] E. Aarts and G. Verhoeven. Chapter g9.5: Genetic local search for the traveling salesman problem. In T. Back, D. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, pages G9.5:1–7. IOP publishing Ltd and Oxford University Press, 1997.
- [2] J. Atkins and W. E. Hart. On the intractability of protein folding with a finite alphabet. *Algorithmica*, 1997. (to appear).
- [3] B. Berger and T. Leight. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. In *RECOMB 98*, 1998. (to appear).
- [4] R. Berretta and P. Moscato. The number partitioning problem: An open challenge for evolutionary computation? In D. Corne, F. Glover, and M. Dorigo, editors, *New Ideas in Optimization*. McGraw-Hill, 1999.
- [5] K. Boese. *Models For Iterative Global Optimization*. PhD thesis, UCLA Computer Science Department, 1996.

- [6] P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, and M. Yannakakis. On the complexity of protein folding. In *RECOMB 98*, 1998. (to appear).
- [7] R. Dawkins. *The Selfish Gene*. Oxford University Press, New York, 1976.
- [8] K. A. Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24:1501, 1985.
- [9] B. Freisleben and P. Merz. A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesman Problems. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pages 616–621. IEEE Press, 1996.
- [10] B. Freisleben and P. Merz. New Genetic Local Search Operators for the Traveling Salesman Problem. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Proceedings of the 4th Conference on Parallel Problem Solving from Nature - PPSN IV*, volume 1141 of *Lecture Notes in Computer Science*, pages 890–900. Springer, 1996.
- [11] W. E. Hart. Adaptive global optimization with local search. *Ph.D. Thesis, University of California, San Diego*, 1994.
- [12] D. Holstein and P. Moscato. Memetic algorithms using guided local search: A case study. In D. Corne, F. Glover, and M. Dorigo, editors, *New Ideas in Optimization*. McGraw-Hill, 1999.
- [13] N. Krasnogor. Two dimensional triangular lattice instances for the hp model. In www.ics.uwe.ac.uk/~natk/INSTANCES/PF/instances.html.
- [14] N. Krasnogor, W. Hart, J. Smith, and D. Pelta. Protein structure prediction with evolutionary algorithms. 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 Kaufman, 1999.
- [15] N. Krasnogor, D. Pelta, P. M. Lopez, P. Mocchiola, and E. de la Canal. Genetic algorithms for the protein folding problem: A critical view. In C. F. E. Alpaydin, editor, *Proceedings of Engineering of Intelligent Systems*. ICSC Academic Press, 1998.
- [16] N. Krasnogor and J. Smith. A memetic algorithm with an adaptive local search. Submitted to PPSN2000. Available from the authors.
- [17] N. Krasnogor and J. Smith. Mafra: A java memetic algorithms framework. In A. Wu, editor, *Workshop Program, Proceedings of the 2000 Genetic and Evolutionary Computation Conference*. Morgan Kaufmann, 2000.
- [18] M. Land. Evolutionary algorithms with local search for combinatorial optimization. *Ph.D. Thesis, University of California, San Diego*, 1998.
- [19] A. Mariano, P. Moscato, and M. Norman. Arbitrarily large planar tsp instances with known optimal tours. *Pesquisa Operacional*, 1995.
- [20] P. Merz and B. Freisleben. Genetic Local Search for the TSP: New Results. In *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, pages 159–164. IEEE Press, 1997.
- [21] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.
- [22] P. Moscato. Memetic algorithms: A short introduction. In D. Corne, F. Glover, and M. Dorigo, editors, *New Ideas in Optimization*. McGraw-Hill, 1999.
- [23] Y. Nagata and S. Kobayashi. Edge assembly crossover: A high-power genetic algorithm for the traveling salesman problem. *Proc. of the 7th Int'l Conf. on GAs*, 1997.
- [24] N. Nunes, K. Chen, and J. Hutchinson. Flexible lattice model to study protein folding. *J. Physical Chemistry*, 100(24):10443–10449, 1996.
- [25] G. Reinelt. Tsplib. In <http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB.html>.
- [26] J. M. Smith. *Shaping Life: Genes, Embryos and Evolution*. Weidenfeld and Nicolson, 1998.
- [27] C. Voudouris and E. Tsang. Guided local search. *European Journal of Operational Research*, 1999.
- [28] J. Watson, C. Ross, V. Eisele, J. Denton, J. Bins, C. Guerra, D. Withley, and A. Howe. The traveling salesrep problem, edge assembly crossover, and 2-opt. *PPSN-V: Parallel Problem Solving From Nature, Proceedings 1998. Lecture Notes in Computer Science*, 1998.