# A Wave Analysis of the Subset Sum Problem*

**Márk Jelasity**
Research Group of Artificial Intelligence
József Attila University, Szeged, Hungary
jelasity@inf.u-szeged.hu

## Abstract

This paper introduces the wave model, a novel approach on analyzing the behavior of GAs. Our aim is to give techniques that have practical relevance and provide tools for improving the performance of the GA or for discovering simple and effective heuristics on certain problem classes. The wave analysis is the process of building wave models of problem instances of a problem class and extracting common features that characterize the problem class in question. A wave model is made of paths which are composed of subsets of the search space (features) that are relevant from the viewpoint of the search. The GA is described as a basicly *sequential* process; a wave motion along the paths that form the wave model. The method is demonstrated via an analysis of the NP-complete subset sum problem. Based on the analysis, problem specific GA modifications and a new heuristic will be suggested that outperform the original GA.

## 1 INTRODUCTION

This paper introduces the wave model, a novel approach on analyzing the behavior of GAs. Our aim is to give techniques that have practical relevance and provide tools for improving the performance of the GA or for discovering simple and effective heuristics on certain problem classes.

This is very important since the models known from the literature are not capable of providing such information. There are measures of problem difficulty such as [Jones et al. 1995], but they tend to be very expensive to calculate and do not provide much more information than the result of running the GA on the given problem. Other approaches suggest features that are responsible for problem difficulty such as deception [Whitley 1991] or having long paths [Horn et al. 1994] but the identification of these features for nontrivial problems is hard and it is not clear, how to improve the performance based on the identified features. Exact models such as Markov chain analysis [Suzuki 1993] are not tractable on nontrivial problems while the wave model is a trade-off between exhaustivity and practical usefulness. Forma analysis [Radcliffe et al. 1994] has similar practical motivations but while it still stands on the ground of the traditional building block hypothesis [Goldberg 1989] the wave analysis is an attempt to shed some light on a rather different aspect of the search process.

In section 2 the basic concepts of the wave analysis will be discussed. In section 3 the practical application of the wave model is demonstrated. The problem class under consideration is the subset sum problem which is NP-complete. After analyzing this problem class, problem specific GA modifications and a new heuristic will be suggested that outperform the original GA. Finally, the results of the paper will be summarized.

## 2 THE WAVE MODEL

First the terminology should be clarified. The *wave analysis* is the process of creating a *wave model* of a fixed objective function or the elements of a characteristic set of functions from a problem class and then extracting the common features of the models. The GA implementation (selection and genetic operators) is also fixed. Thus, a wave model belongs to a problem instance and a GA implementation and the wave analysis is a framework for creating and analyzing such models.

*M. Jelasity (1997) A Wave Analysis of the Subset Sum Problem. In Th. Bäck, ed., *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)*, Morgan Kaufmann, San Francisco, CA, pp89–96

It has to be noted that the analysis is not an automated process. It is a framework that helps creating problem class specific models, but finding a good wave model remains a hard task. The evaluation of the results of the analysis (e.g. the description of the role of the genetic operators) is non-trivial as well. The utility of the approach is not providing trivial methods for gaining information about a problem. Instead, it is a "way of thinking" that makes it possible to learn from the GA how to solve problems, and to develop new, effective and problem class specific heuristics.

As it is widely known, the GA is a very flexible meta-heuristic that is successful on very different problem classes. Models of the GA try to capture the reasons of this flexibility. For example, the oldest, schema based approach suggested, that the search process is nothing else but the identification of 'building blocks' via selection and combining them together via the reproduction operators in an implicitly parallel way. While admitting that in some cases it may be a reasonable model, it is now widely accepted that it is only one of the many strategies a GA can use (see e.g. [Jelasity et al. 1996])

Using the wave analysis we look at the GA as a collection of heuristics and in the case of a given problem class we try to identify the one actually used by the GA. The wave model is *sequential* emphasizing the similarity between the GA and hillclimbing methods. In this framework the GA is in fact a very general and flexible hillclimbing method.

Finally, let us mention that the possibility of generating hard and easy problems with the help of wave models will not be discussed in this paper due to the lack of space though it would be rather interesting. The interested reader will probably form a picture about this issue by the end of this section anyway.

Now, let us fix the notations. Let $S$ be the search space, $f : S \to \mathbb{R}$ the objective function, $C$ the coding space and $g : S \to C$ the injective coding function. For the sake of simplicity, the notation $f(c)$ $(c \in C)$ will be used instead of $f(g^{-1}(c))$. Let $P_0$ be the initial population and $P_i$ the population at step $i$. Let $\overline{f}(P_i)$ be the average function value of the individuals in population $P_i$. The objective function will be maximized.

## 2.1 WAVES

Before introducing the concept of waves, an assumption will be made: $\overline{f}(P_i) \leq \overline{f}(P_j)$ if $i < j$ and the variance of $f$ in the succeeding populations does not increase. This assumption is rather weak since it follows from the properties of the selection mechanisms commonly used in GAs (see e.g.[Blickle et al. 1995]).

A wave needs a space in which it can spread. To construct this space, let us sort the elements of $C$ along a one-dimensional line according to the partial ordering given by $f$. Then, every element in this ordering will be a subset of $C$ with elements having the same function value. Observe that the above assumption means that during the search the population can be looked at as a *wave* that spreads towards the region with the better values. Such a wave is shown in Fig. 1.

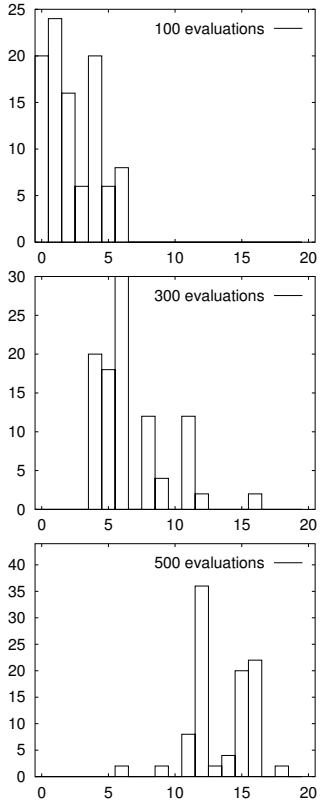


**Fig. 1.** Here $S = [0, 1]$, $f(x) = x$. The population size is 50. Ranking selection and binary encoding were used. The evaluation number is 100, 300 and 500 respectively. The height of the box at point $i$ indicates the proportion of $\text{Prefix}^1_{[i,i]}$ in the population (see Definition 4).

The goal of creating a wave model is to extract the problem specific characteristics of this wave motion. The main method of achieving this goal will be a discretization in terms of characteristic features of $C$ and the result of this will be called a *path*. Paths will be defined in the next section.

## 2.2 PATHS

First, let us define a partial ordering over the subsets of $C$ as it was done in [Vose 1991].

**Definition 1.** *Let* $C_1, C_2 \subset C$. $C_1 < C_2$ *iff* $\max_{c \in C_1} f(c) < \min_{c \in C_2} f(c)$.

The next definition will be the basis of the definition of path.

**Definition 2.** *Let* $C_i \subset C$, $(i = 1, \ldots, k)$. *The sequence* $C_1, \ldots, C_k$ *is an* increasing sequence of features *iff* $C_i < C_j$ *for every* $i < j$.

Every path will be an increasing sequence of features but several restrictions have to be considered. The first and most natural property an increasing sequence must have to be a path is the wave motion property.

**Definition 3.** *An increasing sequence of features* $C_1, \ldots, C_k$ *has the* wave motion property *iff for every* $i$ *Pr(*$C_i \supseteq P_j$ *for some* $j) \approx 1$. *(where Pr() stands for probability and* $P_j$ *is the population at step* $j$*).*

Observe that the succeeding elements of the sequence with the wave motion property has to cover the population one after another because it has been assumed that the average fitness increases during the search and the sequence in question is increasing in the sense of Definition 2. The definition allows us to verify the wave motion property both empirically and mathematically. Figure 2 exemplifies the wave motion property. The definition of the elements of the increasing sequence illustrated in Fig. 2 is the following:

**Definition 4.** *Let* $c \in \{a, b\}^n$. $c$ *has the feature Prefix*$^a_{[i,j]}$ *if the first* $k$ *letter of* $c$ *is* $a$ $(i \leq k \leq j)$ *and if* $k < n$ *then the* $(k+1)^{th}$ *letter of* $c$ *is* $b$.

*Example 1.* Let $C = \{0, 1\}^4$. Then, using the traditional schema notation, Prefix$^1_{[1,2]} = 10{*}{*} \cup 110{*}$, Prefix$^0_{[2,4]} = 001{*} \cup \{0001, 0000\}$.

Let us shed some light on how to read the figures similar to Fig. 2. Every graph in the figures corresponds to a feature. A graph depicts the number of elements in the given generation ($x$-axis) having the feature in question. Instead of averaging the results, the graphs contain a continous line for every experiment performed. For example, Fig. 2 clearly shows that in generation 10 Prefix$^1_{[0,5]}$ is almost not represented in most of the experiments, Prefix$^1_{[6,12]}$ dominates the generation i.e. the wave is here in generation 10 and Prefix$^1_{[13,20]}$ starts gaining strength.

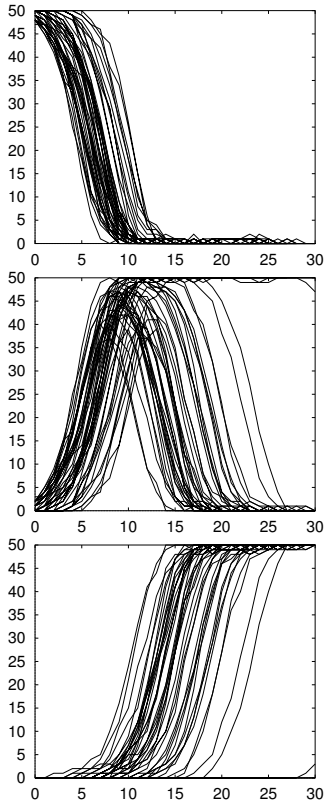


**Fig. 2.** Here $S = [0, 1]$, $f(x) = x$. The population size was 50. Ranking selection and 20-bit binary encoding were used. The evaluation number was 1000. 50 independent runs were performed. The number of representatives of the features Prefix$^1_{[0,5]}$, Prefix$^1_{[6,12]}$ and Prefix$^1_{[13,20]}$ are shown as the function of generation index for every run.

At this point a natural question arises: can we accept an increasing sequence as a model of the GA if the sequence in question shows the wave motion property. The answer is certainly no. The problem is that if $P_i \subset A$ holds for some feature $A$ and for a population $P_i$ then $P_i \subset B$ will also be true for any $B \supset A$. To overcome this difficulty, it has to be required that every element of the increasing sequence of features has to be *minimal* in the sense of the next definition:

**Definition 5.** *An element of an increasing sequence with the wave motion property* $C_i$ *is* minimal *if the replacing of* $C_i$ *with any of its subsets results in a new sequence that does not have the wave motion property anymore.*

Now the definition of a *path* can be given.

**Definition 6.** *An increasing sequence of features is a* path *if it has the wave motion property and every element is minimal in it.*

### 2.3 PATH DECOMPOSITION

Definition 6 is still not sufficient for our purposes; some refinements have to be made. It may very well happen that a path says little about the process inside the GA and cannot be a basis of improving the performance of the search. The problem is connected with the multimodality of the objective function. To shed some light on this issue, let us consider the example shown in Fig. 3. Though it is a path,

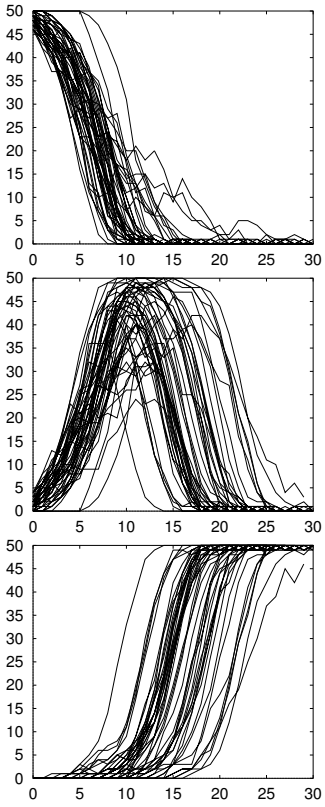

**Fig. 3.** Here $S = [0, 1]$, $f(x) = |x-0.5|$. The population size was 50. Ranking selection and 20-bit binary encoding were used. Only 1-point crossover was used with a probability of 1. The evaluation number was 1000. 50 independent runs were performed. The number of representatives of the features $\text{Prefix}^1_{[1,5]} \cup \text{Prefix}^0_{[1,5]}$, $\text{Prefix}^1_{[6,12]} \cup \text{Prefix}^0_{[6,12]}$ and $\text{Prefix}^1_{[13,20]} \cup \text{Prefix}^0_{[13,20]}$ are shown as the function of generation index for every run.

it is clear that if for the starting population $P_0 \subset \text{Prefix}^1_{[1,5]}$ holds then with the given settings no solutions will be generated that would start with a 0 and in fact the search will be identical with the earlier example shown in Fig. 2 so the sequence $\text{Prefix}^1_{[1,5]}$, $\text{Prefix}^1_{[6,12]}$, $\text{Prefix}^1_{[13,20]}$ is a path. Similarly, its 0-prefixed counterpart is a path as well. The above comments make it clear that the path in question has some kind of structure and the information about this structure is essential from the viewpoint of a good model. The above phenomenon motivates the next definition.

**Definition 7.** *A path $C_1, \ldots, C_k$ is* complex *iff there are two paths $B_1, \ldots, B_k$ and $A_1, \ldots, A_k$ such that $B_i \cap A_i = \emptyset$ and $B_i \cup A_i = C_i$ $(i = 1, \ldots, k)$. If a path is not complex then it is* simple.

Now the definition of the *wave model* can be given.

**Definition 8.** *A* wave model *of the search performed by an implementation of the GA on a given objective function is a set of simple paths such that for every method used for generating the initial population $P_0$ there is exactly one path in which the first feature $C_1$ covers $P_0$ with a probability approaching 1 (Pr($C_1 \supset P_0$) $\approx 1$).*

This definition of the wave model is very simple and could be refined in several ways. For example it says nothing about the relation of different paths or other possible types of decompositions of paths. However, for the present discussion it suffices since the focus is on the empirical results of section 3.

## 3 THE SUBSET SUM PROBLEM

In this section we demonstrate the wave analysis using the subset sum problem which is NP-complete. Then, using the wave model, the performance of the GA will be improved and a heuristic will also be given that outperforms the original GA.

### 3.1 PROBLEM DESCRIPTION AND REPRESENTATION

In the case of the subset sum problem we are given a set $W = \{w_1, w_2, \ldots, w_n\}$ of $n$ integers and a large integer $M$. We would like to find a $V \subseteq W$ such that the sum of the elements in $V$ is closest to, without exceeding, $M$. This problem is NP-complete. Let us denote the sum of the elements in $W$ by $SW$.

We created our problem instances in a similar way to the method used in [Khuri et al. 1993]. The size of $W$ was set to 100 and the elements of $W$ were drawn randomly with a uniform distribution from the interval $[0, 10^4]$ instead of $[0, 10^3]$ (as was done in [Khuri et al. 1993]) to obtain larger variance. According to the preliminary experiments, the larger variance of $W$ results in harder problem instances. Five problem instances were generated (SUB1, SUB2, SUB3, SUB4 and SUB5). Since the value of $M$ seemed to be interesting during the preliminary experiments, $M$-s was set in a different way for all the five instances. We set $M_i$ ($M$ corresponding to the $i^{th}$ problem

instance SUB$i$) to the closest integer to $SW_i \cdot i/9$ where $SW_i$ is the $SW$ corresponding to SUB$i$.[1] (It should be noted that exact solutions do exist for the examined problem instances.)

We used the same coding and objective function as suggested in [Khuri et al. 1993]. $C$ was $\{0,1\}^{100}$. If $x \in C$ ($x = (x_1, x_2, \ldots, x_{100})$), then let $P(x) = \sum_{i=1}^{100} x_i w_i$, and then

$$-f(x) = a(M - P(x)) + (1-a)P(x)$$

where $a = 1$ when $x$ is feasible (i.e. $M - P(e) \geq 0$) and $a = 0$ otherwise.

### 3.2 WAVE ANALYSIS

The experiments were performed with GENESIS [Grefenstette 1984]. The selection type was ranking selection. The operators were 1-point crossover and traditional mutation. The probabilities of the operators are 1 and $0.003$ if not otherwise stated. The population size was 100 and the number of evaluations was 5000 in every experiment. The initial populations were generated by a uniform random sampling of $C$.

Before giving the analysis an important issue has to be discussed: the methods for identifying the features that would form the paths of the wave model. In general, it is a tough problem and requires a lot of work. In fact, it needs a scientific research: making a hypothesis, verifying it doing experiments with the GA, improving the hypothesis and so on. The difficulty is hidden in the fact that the set of possible features for given configurations of the GA is very large and mostly *undiscovered*. Schemta form only a (maybe small) subset of this collection of features. For example, the features that will arise in this work are fairly independent of schemata and other examples are given in [Jelasity et al. 1996]. It is very likely that any automation of this feature-finding process (if possible) would involve very powerful and intelligent computational methods. The question is: is it worth doing the above research? This section implies that the answer is yes. There is hope that as a result of giving a wave model of a problem, we can extract common features of the whole problem class that makes it possible to improve the performance of the GA or even to develop effective problem class specific heuristics.

Now let us see the wave analysis of the subset sum problem. Experimenting with the GA, it has been found that on every problem instances the search has two phases: the distribution optimization phase and the hillclimbing phase.

[1] Instances with an $M > SW/2$ have an 'almost' equivalent problem with an $M' = SW - M$. 'Almost' equivalent, because of the asymmetric construction of the objective function.

#### 3.2.1 Distribution optimization

This phase is connected to the size of $M$. The method for generating the initial population has a special bias regarding the number of bits. This factor has a gaussian distribution with a mean value of 50 (the half of the string-length). This means that most of the elements in $P_0$ have approximately 50 1s so the expected value of the fitness function is $SW/2$. If $M$ is smaller than $SW/2$ then the initial population can be expected to have a poor performance. Distribution optimization means that the GA alters the distribution and the number of 1s to a better configuration. This phenomenon is the most characteristic in the case of SUB1 so we will concentrate on this problem instance here. To construct a path for SUB1 let us first define a feature:

**Definition 9.** *A $c \in C$ has the feature $L_{[i,j]}$ if the subset defined by c contains $k$ of the largest 50 elements of $W$ and $i \leq k \leq j$.*

Then, we claim that the sequence $L_{[15,30]}$, $L_{[5,14]}$, $L_{[0,4]}$ is a path. The mathematical considerations implying that the above sequence is increasing with a high probability (w.r.t. the samples taken by the succeeding populations) are straightforward and elemental and therefore omitted. The empirical results shown in Fig. 4 imply the wave property. The minimality and simplicity of the path are also trivial if considering the definitions of these properties (see section 2).

Another argument beside this model is that it predicts[2] that the high mutation probability which has a bias towards the initial distribution of bits in the solutions detaining the wave motion of the above path will decrease the performance. Experimental results justify the prediction (see Fig. 5 and NAIV-M in Table 2).

#### 3.2.2 The Hillclimbing Phase

This phase begins when the optimization of the bit distribution in the solutions has been performed. The model of this second phase does not share the linear style of the first phase. On the contrary, we suggest that there is an enormous number of paths in the model of this phase that are built of relatively small sets and are highly problem instance specific. This is why this phase is called the hillclimbing phase; such path structure calls for a hillclimbing strategy. This claim is supported by section 3.3.

There are several arguments that support our suggestion regarding the path structure of this phase. First, every run

[2] Prediction is possible because the path under consideration covers the whole search space and therefore does not allow any other paths to exist.

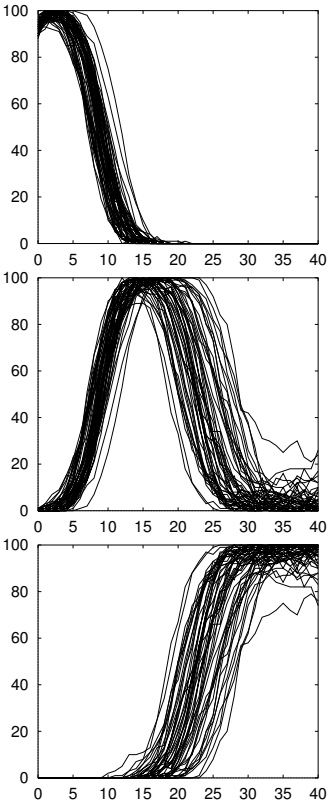

**Fig. 4.** The number of representatives of $L_{[15,30]}$, $L_{[5,14]}$ and $L_{[0,4]}$ respectively. The results of 50 runs are shown as a function of generation index.

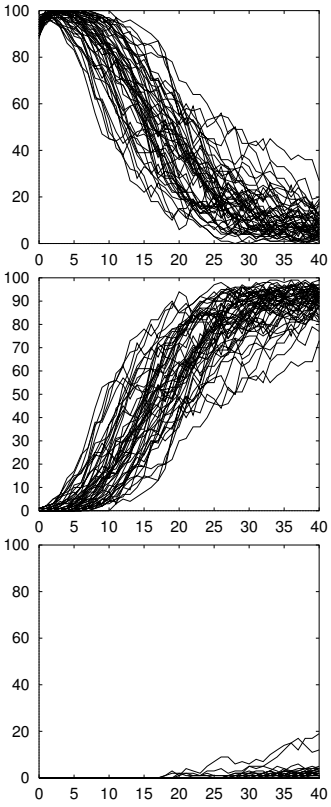

**Fig. 5.** The proportions of $L_{[15,30]}$, $L_{[5,14]}$ and $L_{[0,4]}$ respectively. The probability of mutation is increased to $0.06$. The results of 50 runs are shown as a function of generation index.

on every problem resulted in a different solution that are considerably far from each other (see Table 1). The many optimal solutions found do not seem to have any common feature except the bit distribution. Results of coding theory [Lint 1992] also support that a great number of paths can exist without interfering with each other. As it was shown in [Jelasity et al. 1995], GAS, a GA with a special niching technique supporting the separate handling of different local optima outperformed the standard GA on this problem. Finally, the modifications of the GA that were made using this hypothesis were successful as it will be seen in section 3.3.

### 3.2.3 The Wave Model

As the mindful reader has observed already, no exact wave models have been given for any of the problem instances under consideration. Since the aim of the wave analysis is to extract characteristic features of whole problem classes, the problem instance specific details (such as the exact path structure of the second phase for a given problem instance) are not important. What was given is a general characterization of the search on an arbitrary problem instance of a class of the subset sum problem.

## 3.3 APPLICATION OF THE RESULTS

As it has been suggested, the search has two phases: the bit distribution optimization phase and the hillclimbing phase. It will be shown that both require extra computational effort that can be saved. In the following, the modified algorithms will be described.

OPTDISTR**, distribution optimization.** This phase can be totally eliminated by explicitly ensuring that the bit distribution is optimal from the very beginning of the search w.r.t the bias of the population initialization procedure. This was done by modifying the problem instances.[3] For a problem instance SUB$i$ from the base set $W_i$ the $k_i$ largest elements have been deleted where $k_i$ was such that the sum of the remaining elements of $W_i$ was the closest to $2M_i$. A solu-

[3] The algorithm of the modification is independent from the problem instances so can be looked at as a problem class specific modification of the GA.

| | Hamming distance | | | |
|---|---|---|---|---|
| | min. | max. | average | variance |
| SUB1 | 13 | 69 | 47.56 | 91.89 |
| SUB2 | 24 | 63 | 47.55 | 50.5 |
| SUB3 | 34 | 65 | 49.91 | 26.2 |
| SUB4 | 34 | 68 | 49.71 | 26.77 |
| SUB5 | 35 | 64 | 49.97 | 25.21 |

**Table 1.** The values correspond to sets of optimal solutions of problem instances found during all the experiments icluding the ones with the modifications of the GA (section 3.3). The minimum, the maximum, the average and the variance of the Hamming distences of pairs from these sets are shown.

tion of a modified problem instance naturally defines a solution of the original problem instance with the same function value.

All the following algorithms in this section use these modified problem instances.

5X1000**, hillclimbing phase.** According to our model, there are a lot of paths in this phase. Since they are rather far from each other (see 1) and do not seem to show any common structure it was assumed that it would be a good idea process them separately. Therefore the population size was reduced to 2 and the GA was run 5 times with 1000 evaluations in each to ensure that only one path is processed at a time. Then, the best solution was picked as a result. The only operator was mutation with a probability of 0.06. Note that this algorithm is rather similar to – though more flexible than – the stochastic hillclimber.

HEUR**, a heuristic.** To examine the effect of the optimal bit distribution a heuristic has been introduced which simply generated 5000 random individuals on the modified problem instances. This method is in fact equivalent to generating an initial population with 5000 elements.

### 3.3.1 Evaluation

It can be seen that the optimal bit distribution is essential; even the random search (HEUR) performed well though only the bit distribution was optimized.

The application of the information about the hillclimbing phase was useful as well. 5X1000 had the best average performance on almost every problem instance especially on SUB5 which is the hardest (the largest) problem instance since the smallest set is subtracted from $W_5$ due to the bit distribution optimization. Note that no fine tuning of the parameters have been performed to adapt the method to smaller problems. Table 2 clearly shows that the model has practical relevance.

## 4 SUMMARY

In this paper the wave analysis of GAs has been described. The wave analysis is the process of building wave models of problem instances of a problem class and extracting common features that characterize the problem class in question. A wave model is made of paths which are composed of subsets of the search space (features) that are relevant from the viewpoint of the search. The GA is described as a basicly *sequential* process; a wave motion along the paths that form the wave model.

The above mentioned features include but are not at all limited to schemata. In fact there are many that are independent of schemata such as those involved in the wave analysis of the subset sum problem presented in this paper. Using this analysis, modifications of the naiv GA has been suggested that outperformed the original algorithm on the subset sum problem class.

## References


[Grefenstette 1984] J.J. Grefenstette (1984) GENESIS: A System for Using Genetic Search Procedures, in *Proceedings of the 1984 Conference on Intelligent Systems and Machines*, (pp161-165).

[Goldberg 1989] D. E. Goldberg (1989), *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, ISBN 0-201-15767-5.

[Vose 1991] M.D. Vose (1991) Generalizing the Notion of Schemata in Genetic Algorithms, *Artificial Intelligence*, 50:385-396.

[Whitley 1991] L.D. Whitley (1991) Fundamental Principles of Deception in Genetic Algorithms, in *The Proceedings of FOGA'91*, Morgan Kaufmann.

[Lint 1992] J.H. van Lint (1992) *Introduction to Coding Theory.*, Springer-Verlag.

[Khuri et al. 1993] S. Khuri, T. Bäck, J. Heitkötter (1993), An Evolutionary Approach to Combinatorial Optimization Problems, in *The Proceedings of CSC'94*.

[Suzuki 1993] J. Suzuki (1993) A Markov Chain Analysis on a Genetic Algorithm, in *The Proceedings of ICGA'93* pp146-153.

[Horn et al. 1994] J. Horn, D.E. Goldberg, K. Deb (1994) Long Path Problems, in *The Proceedings of PPSN III*, Springer.

[Radcliffe et al. 1994] N.J. Radcliffe, P.D. Surry (1994) Fitness Variance of Formae and Performance Prediction, in L.D. Whitley and M.D. Vose editors, *Foundations of Genetic Algorithms III*, Morgan Kaufmann (San Mateo, CA) pp51-72.

[Blickle et al. 1995] T. Blickle, L. Thiele (1995) *A Comparison of Selection Schemes used in Genetic Algorithms* (2. Edition), Technical Report, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich.

[Jelasity et al. 1995] M. Jelasity, J. Dombi (1995) GAS, an Approach on Modeling Species in Genetic Algorithms, in *The proceedings of EA'95*.

[Jones et al. 1995] T.Jones, S. Forrest (1995) Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms, Submitted to ICGA'95 (Jan. 1995).

| | NAIV | | NAIV-M | | OPTDISTR | | OPTDISTR-M | | 5X1000 | | HEUR | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | #opt | average | #opt | average | #opt | average | #opt | average | #opt | average | #opt | average |
| SUB1 | 4 | -8.0 | 0 | -6136.0 | 14 | -5.26 | 12 | -2.72 | 5 | -5.8 | 15 | -3.48 |
| SUB2 | 5 | -7.64 | 0 | -186.8 | 11 | -3.92 | 9 | -4.34 | 9 | -4.0 | 9 | -7.96 |
| SUB3 | 3 | -10.5 | 2 | -20.32 | 9 | -3.98 | 9 | -4.8 | 9 | -3.35 | 4 | -8.6 |
| SUB4 | 5 | -7.94 | 6 | -8.62 | 7 | -5.5 | 10 | -5.94 | 11 | -4.0 | 4 | -13.8 |
| SUB5 | 5 | -9.6 | 8 | -6.12 | 9 | -6.7 | 5 | -7.72 | 14 | -3.76 | 2 | -13.94 |

**Table 2.** The methods used are described in the text. NAIV is the GA used in section 3.2. '-M' means that only mutation was used with a probability of 0.06. The values correspond to the result of 50 independent runs. The number of optimal solutions found and the average of the results of the runs are shown.

[Jelasity et al. 1996] M. Jelasity, J. Dombi (1996) Implicit Formae in Genetic Algorithms, in *The proceedings of PPSN IV* pp154-163 LNCS 1141, Springer.