The Effect of Cost Distributions on Evolutionary Optimization Algorithms

César Galindo-Legaria
Microsoft Corp.
One Microsoft Way
Redmond, WA 98052
USA
cesarg@microsoft.com

Florian Waas[⋆]
CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands
flw@cwi.nl

Abstract

According to the No-Free-Lunch theorems of Wolpert and Macready, we cannot expect one generic optimization technique to outperform others on average [WM97]. For every optimization technique there exist “easy” and “hard” problems. However, little is known as to what criteria determine the success of an optimization technique.

In this paper, we consider this question from the evolutionary computing point of view. We use cost distributions, i.e., the frequencies of the objective function’s values occurring in the search spaces, to devise a classification of optimization problems. Unlike fitness landscapes, the cost distribution is truly problem intrinsic rathern than part of an algorithmic solution.

Based on the characteristic cost distribution of a problem, our model helps to predict what components of an evolutionary algorithm are most relevant (e.g., initialization, mutation), and what is the expected overall performance. We validate the model through experiments on three problems: Set Partitioning, Knapsack, and Traveling Salesman.

1 INTRODUCTION

Evolutionary algorithms have been shown to be very successful for a variety of optimization problems, for example timetabling and scheduling. However, there are a number of other problems, like Traveling Salesman, that have been much more difficult to tackle using the same techniques.

There is some intrinsic property of a problem that makes it more or less suitable to be tackled by evolutionary algorithms. In the past, researchers have observed three major classes of NP-complete optimization problems:

Easy. Near-optimal or even optimal solutions can be found without major difficulties. In this case, sophisticated mutation or crossover operations have only marginal impact on the quality of the result. In a sense, these problems are “too easy” for the evolutionary framework. Simpler optimization algorithms like Hill Climbing often outperform evoluationary techniques in that they find results of comparable quality much quicker. An example of this kind of problem is Set Partitioning.

Adequate. A large class of problems belong to this class where evolutionary algorithms excel, often outperforming other optimization strategies significantly. One example of this kind of problem is Knapsack.

Hard. Problems like the Traveling Salesman Problem pose particular difficulties to evoluationary optimization. In this case sophisticated, problem-specific tuning is necessary to obtain acceptable results. In contrast to *easy* problems, evolutionary optimization seldom finds optimal solutions. Most interestingly, this particular difficulty appears to be of a general nature, independent of the type of evolutionary optimization used or topology defined in the search space.

In this paper we investigate the role of *cost distributions* of optimization problems. By this we mean the frequencies of all occurring values of the objective function throughout the entire search space. It provides basic statistical information on average cost of a random solution, and concentration of good or bad solutions. Cost distributions are independent of any notion of adjacency, proximity, or neighborhood of solutions, defined for example as an algorithmic transformation between solutions. Rather, given a problem instance,

[⋆] Author’s current address: Microsoft Corp, Redmond, WA 98052, USA

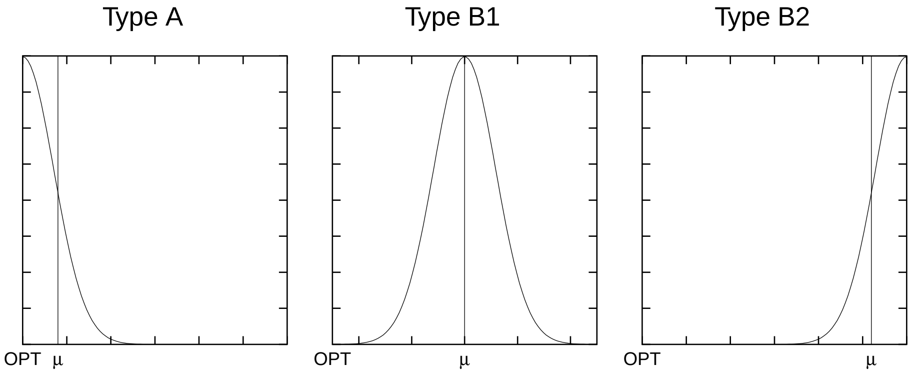

Figure 1: *Basic types of distributions (qualitatively). Optimum denoted by OPT, mean by μ;*

its cost distribution underlies *all* possible topologies a search algorithm can define on the search space.

Through a large number of experiments, we have seen that cost distributions appear to be very characteristic for different optimization problems [Waa99, WGL00a]: Different instances of a given problem exhibit cost distributions which are of similar quality. The variation within a problem is gradual and limited.

We present here a broad classification of cost distributions, based on surveying the literature and own experimental observations. We discuss how each building block of the evolutionary framework (e.g., initialization, mutation) is affected by the cost distribution, and what is the overall impact on the optimization algorithm. Our model helps explain earlier results on the behavior of evolutionary optimization. Also, we believe our analysis of individual components can aid in tailoring the general evolutionary framework to specific problems, and our general model will be useful to predict how amenable are new problems for treatment using evolutionary techniques.

2 PARAMETERS OF SEARCH SPACE

Since the introduction of general search algorithms, significant effort has been devoted to characterize the *search space*—i.e., the set of all possible solutions—and its influence on the search algorithms. In this section we discuss cost distributions and how they influence topological models.

2.1 COST DISTRIBUTIONS

A cost distribution captures the frequencies of cost values—i.e., values of the objective function—in the complete search space. [1] For any particular cost c, the distribution indicates the number of feasible solutions in the space whose cost is c. This information is the basis to answer questions such as: Are there "many solutions" close to the optimum?

When a space of solutions is too large to be enumerated, the general shape of the cost distribution can be approximated by uniform random sampling of the space (or quasi-random sampling like random walks, when uniform sampling is too hard).

Cost distributions are independent of the algorithm used to tackle the problem and are an invariant property of the particular problem instance. No matter whether a topology is defined at a later stage, the cost values and their frequencies are not altered.

What makes cost distributions such an important instrument is the possibility to analyze concentrations of cost values in the search space. We are in particular interested in the distance between the optimum and the bulk of solutions. Without loss of generality we assume a minimization problem. The question central to our further considerations is therefore:

> *Is the bulk of solutions close to the optimum or is the optimum an outlier with respect to the distribution?*

In large test series with different optimization problems, we have observed two basic types A and B of cost distributions, the second of which comes as two sub-types B1 and B2. In Figure 1, these distributions are shown qualitatively.

In problems with type-A distribution, the bulk of solu-

[1] We prefer the term *cost* over *fitness*, because sometimes fitness is intended as a relative measure (see e.g., [ZT99]). Instead, cost refers to the absolute value of the objective function.

tions is very close to the optimal costs, i.e., there are many optimal or near-optimal solutions in the search space. The optimum can even have highest frequency of all solutions (see Sec. 4.1). Note that this cost-wise proximity does not imply any neighborhood or topology, but simply indicates that many different solutions with similar cost exist.

In problems with type-B distribution, the bulk of solutions is of distinctly different cost than the optimum. We can distinguish the sub-types B1 where the mean is at a moderate distance of the optimum, and B2 where the bulk of solutions is far away from the optimum, i.e., the optimum is an outlier and near-optimal solutions are rare.

Note, the actual shape of the distribution–symmetry, skew, etc.–is of little relevance. The concentration of solutions relative to the optimum will be important for the further analysis. In practical problems cost distributions will likely show disturbances, and certainly not all problems can be assigned exactly to one type but may be in between two types. Yet, we can identify clear trends and the results of our analysis can be interpolated as necessary.

2.2 WHAT FITNESS LANDSCAPE?

Models for the *topology* or *landscape* of the space are powerful tools to interpret certain effects occurring with optimization algorithms (see e.g., [Kau93]). However, unlike the cost distribution, the space landscape is not intrinsic to the problem, and completely different topologies can be defined for a given space.

For instance, consider the Traveling Salesman Problem, where the shortest tour via a number of cities is sought. Let us define two different notions of neighborhood N1 and N2. Two tours are neighbored if one can be transformed into the other by

N1: exchanging two subsequently visited cities;

N2: exchanging *any* two cities;

Figure 2 illustrates the consequences with the *optimal* tour and possible neighbors according to the two different neighborhood relations. Whereas neighbors under N1 are of very similar cost, neighbors under N2 can be of higher differences in costs. Moreover, N2 is a super set of N1, i.e., neighbors in N1 are also neighbors in N2 but not conversely.

Both N1 and N2 induce a topology, thus a landscape, on the search space. One appears relatively smooth (N1), the other rugged (N2). Yet they are both defined on the same problem. Neither of the two landscapes is intrinsic to the problem, and neither could be claimed to be “the natural” landscape. A number of other neighborhood relations for this problem have been described in literature; for a survey on neighborhood defining transformations see for instance [FJMO95].

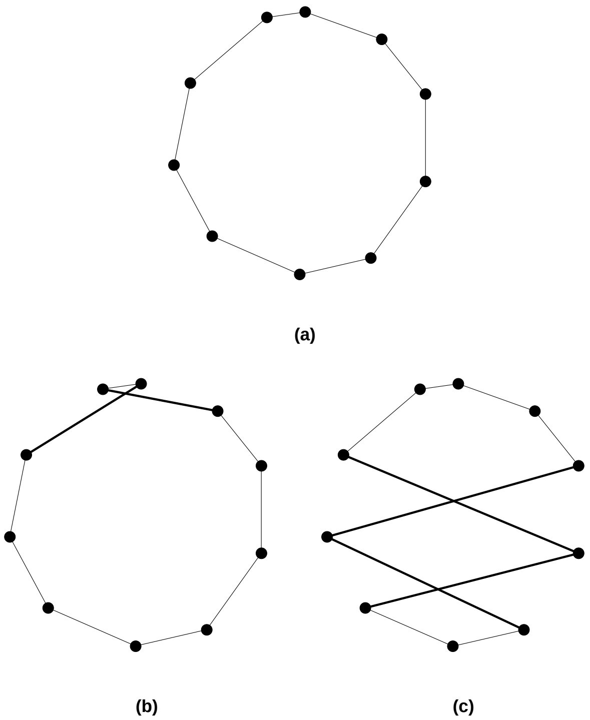

Figure 2: *Alternative tours for a Traveling Salesman Problem; (a) optimal tour, (b) tour neighbored to optimal tour under N1, (b) under N2*

In our view, the space landscape is part of the approach to solve a problem. In contrast, the cost distribution is an intrinsic attribute of the problem. A main claim of this paper is that useful insight can be gained from considering the cost distribution alone, and that such insight holds regardless of the specific topology imposed on the space.

3 PRINCIPLES OF EVOLUTIONARY ALGORITHMS

The notion of evolutionary computing is fairly flexible, comprising a large variety of algorithms and techniques. Frameworks as for instance presented in [Gol89, Mit96] are capable of simulating other algorithms that are commonly not considered evolutionary, like Random Sampling or Simulated Annealing [Bäc96]. On the other hand there are several generic elements that are agreed to be characteristic for an evo-

	Type-A	Type-B1	Type-B2
Initialization	$\oplus$	$\odot$	$\odot$
Recombination	$\oplus$	$\oplus$	$\oplus\oplus$
Mutation	$\oplus$	$\odot$	$\ominus$
Restarts	$\oplus$	$\odot\ominus$	$\odot\ominus$
Overall	easy	adequate	hard

($\oplus$)$\oplus$ = (strong) positive influence, $\odot$ = no influence, $\ominus$ = negative influence

Table 1: *Importance of components of evolutionary algorithms with respect to the cost distribution*

lutionary algorithm. In our analysis, we first sketch this generic key elements and scrutinize the impact of cost distributions on those components. In particular, we investigate to what degree the single components use randomly selected solutions. Random sampling, uniform or biased, on—possibly restricted—sets of solutions is the very nucleus of all randomized optimization algorithms including evolutionary techniques.

Starting with a randomly generated initial population, generations are repeatedly derived by selecting a set of parents, generating the offspring by *recombination*, introducing a certain random distortion in form of *mutation*, and subjecting all individuals to a *selection* process. The algorithm terminates as soon as a certain stopping criterion—e.g., timeout, maximum number of individuals reached, or no improvement over a certain number of generations—is fulfilled. In every generation, all individuals are checked for their *fitness*, i.e., their costs, not only for the selection of the next generation but also to keep track of the best individual found so far. Simulating the natural evolutionary process the algorithm achieves a gradual improvement concentrating on well suited individuals by selection and the production of closely related offspring.

Initialization. The influence of the cost distribution on the initial phase is significant as *initializing* directly translates to *sampling*. Note, that sampling here does not necessarily mean uniform sampling.

For a type-A distribution the probability to find already near-optimal solutions in the initial sample is high. In other words, the subsequent optimization phase cannot improve the initially found solutions substantially. The probability that high quality solutions are included in the initial solution depend further on the size of the population: *very* small populations may differ enormously in quality.

In case of a type-B distribution, the initialization's role is less important, depending on the distance of the cost of the optimal solution from the average cost. The sampled initial individuals are of comparable, distinctly sub-optimal quality. In type-B2 problems, the initialization produces only results of constant but low quality. As opposed to the previous case, the size of the population does not affect its quality—the probability to sample a near-optimal solution is virtually zero.

Recombination. Implementing a mating between two individuals results in a *random* solution which consists of parts of its ancestor.

In the case of the type-A distribution sophistication is usually of limited use only as there are plenty of solutions in the close vicinity. However, if there are too many close relatives, guiding the recombination process becomes also more difficult.

The less solutions with similar costs to their ancestors there are, the more astray—i.e., in direction of the average cost—the recombination may lead. More sophisticated algorithms are necessary to avoid a fall back to the bulk of solutions in case of a type-B2 distribution.

Mutation. In case of a type-A distribution, mutation can be most fruitful as the odds to improve by random alteration are high.

For a type-B distribution, the probability to achieve an immediate improvement by mutation is very small but mutation is still useful to avoid undue concentration of certain properties among the individuals.

Restarts. Evolutionary algorithms, mimicking the natural evolutionary process are characterized by convergence, i.e., the overall fitness of the consecutive generations increases—although it is not necessarily monotonic. For simplified models of those algorithms, the convergence of the optimum as a limit, provided an infinite running time, has been proven (see e.g., [Bäc96]). Similar facts are known for algorithms like Simulated Annealing. However, depending on the cost distribution, evolutionary algorithms can very well profit from restarting, simply because of the cost distribution's influence on the initialization.In case of a type-A distribution, the impact of re-runs may greatly improve the results, whereas in a type-B1 scenario, restarts do not make much of a difference. The influence is even weaker in case of a type-B2 problem. With type-B distributions, the results usually do not justify the higher costs in terms of running time.

In Table 1, the basic tendencies of influence are summarized. The three types of cost distributions directly suggest three classes of difficulty—from an evolutionary algorithm point of view. Type-A is the easiest, where all components but recombination are positively influenced by the distribution. The impact on Type-B1 problems is fairly balanced; in Type-B2 problems negative influences dominate.

4 CASE STUDY

To corroborate our analysis, we scrutinize three representative and well-understood NP-complete optimization problems: *Set Partioning*, *Knapsack*, and *Traveling Salesman* Problem (see e.g. [GJ79]). The three problems are archetypical optimization problems which have been receiving great attention ever since their inception.

As a preliminary study to the experiments presented below, we scrutinized the occurring cost distributions with respect to the variance among instances of the same problem. For all three types of problems we varied all available parameters like size, type and parameters of the distributions of weights or coordinates etc. and determined the cost distributions with large uniform samples.

We found the cost distributions of our three problems converge very quickly (i.e. for virtually all non-trivial problem sizes) to their anticipated analytical continuous approximation, according to the Central Limit Theorem. This statistical analysis, based on the structure of the problem and its cost function, is presented in [Waa99]. For TSP, we have also determined the cost distributions of all problems given in the standard benchmark library TSPLIB [Rei91, WGL00b], and found them to match the same characteristic shape.

In this section we will present data taken from a larger series of experiments in which we scrutinized the effect of the cost distributions. To ensure the results of the experiments are comparable across the different problems, we implemented a generic framework which provides a uniform way of controling common parameters like the ratio of individuals generated by recombination to the those stemming from mutation etc. All experiments below were conducted using 250 individuals per generation; the optimization was terminated after 1000 generations.

While writing our own framework guarantees a fair analysis, we compared our findings with results in the literature verifying the generality of our observations.

4.1 TYPE-A

Set or number partioning is a typical representative of this class. A set S of numbers is to be partitioned into 2 subsets S_1 and S_2 such that $S_1 \cup S_2 = S$, and the difference of the sums

$$|\sum_{s \in S_1} s - \sum_{s \in S_2} s|$$

is minimal.

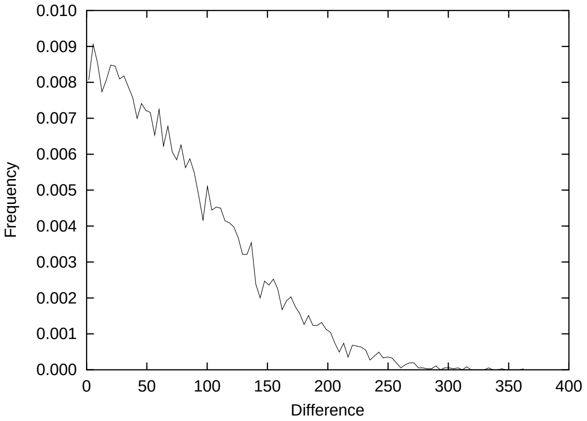

Figure 3: *Set Partitioning: Cost distribution for problem instance of size 150*

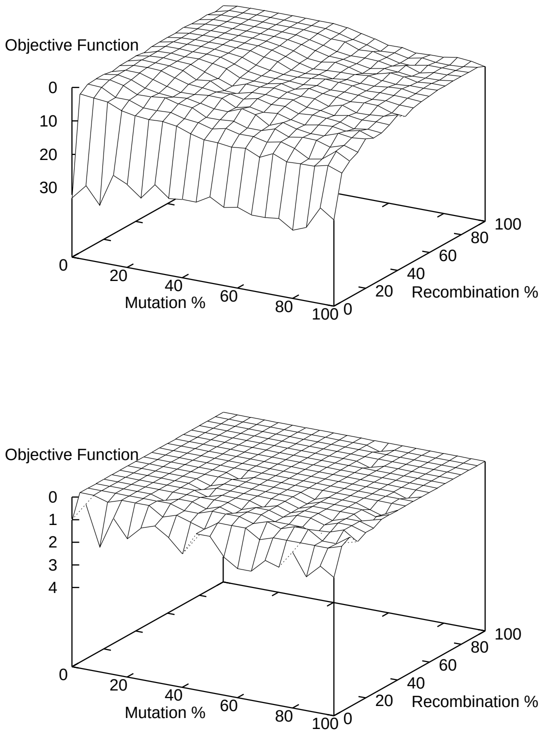

Figure 4: *Set Partitioning: Quality of optimization result as function of mutation and recombination ratio; z-scale inverted, smaller values indicate better results (optimum at 0). Population size 100 (above), 1000 (below)*

Figure 3 shows the cost distribution of an instance with 100 elements taken from a Gamma distribution. The original distribution of numbers decreases in significance with increasing size of the set. For an analytical model, we refer the reader to [WGL00b]. The cost distribution is characterized by optimal and near-optimal costs appearing with the highest frequencies. Even plain random sampling algorithms are guaranteed to find near-optimal solutions [KKLO86], hill climber and other multi-start algorithms that do not deploy highly sophisticated techniques, achieve excellent results within extremely short running time.

Evolutionary algorithms find results of similar quality but require longer running times. With this kind of distribution, the size of the initial population is critical to the stability of the optimization, i.e., using a population size of 1000 almost certainly contains an optimal or near-optimal solution; the quality of small populations may differ significantly, so that using a tight time limit and re-starting the algorithm a couple of times may improve the results significantly in case of small populations.

We implemented a genetic algorithm for set partitioning using the standard string encoding. Figure 4 shows the dependencies between the quality of the optimization result and recombination and mutation ratio respectively. Besides the ratio of mutation and recombination, we also varied the size of the population.

- The figure underlines the importance of the initialization (value for (0,0) represents plain random sampling): Using a large populations we obtain near-optimal results without recombination or mutation
- We achieve (near-)optimal results easily for almost any configuration
- No sophistication is needed when defining the recombination operation

4.2 TYPE-B1

The class of type-B1 distributions comprises, among others, a large variety of scheduling, timetabling, assignment problems, and *Knapsack Problems* on which we focus here.

The problems definition is as follows: Given a number of items—each has a profit and a weight associated with it—, a (sub-)set of items is sought such that the total weight does not exceed a given bound but the sum of profits is maximal (see e.g., [GJ79]). We inverted the values to turn the maximization problem in one of finding the minimum. In Figure 5, the cost distribution of an instance consisting of 150 items is shown. The values of both weight and profit of the single items were chosen as random numbers between 10 and 100. The capacity of the knapsack was chosen as half the total weight of all items. Such assumptions are common in the literature [ZT99]—in particular, this configuration follows the example of [MT90].

The effect of this distribution on genetic search is twofold: The sampling of an initial population does not contain high quality solutions. Also, the random sampling component within cross over and mutation is limited—the probability to sample a near optimal solution is practically zero. On the other hand, the optima are not too far away from the bulk of solutions. Genetic algorithms are known as a suitable and very successful optimization technique for this kind of cost distributions.

Figure 6 shows the experimental results obtained with an implementation using standard string encoding of individuals.

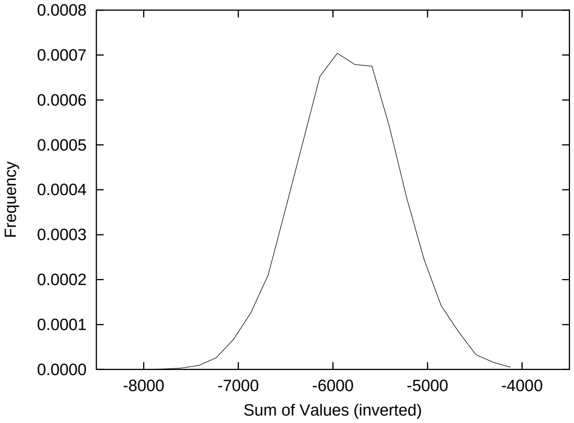

Figure 5: *Knapsack Problem: Cost distribution for problem instance of size 150; x-scale negated; (optimum at -8386)*

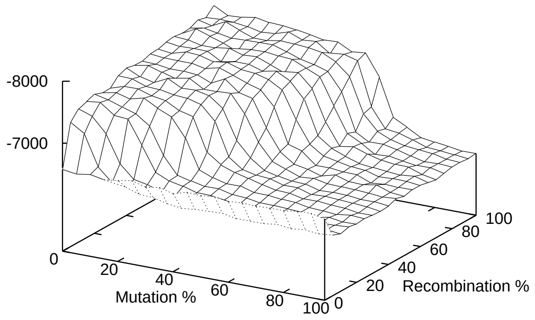

Figure 6: *Knapsack Problem: Quality of optimization result as function of mutation and recombination ratio*

- In contrast to type-A problems, the population size is of little importance (we omitted figures for experiments with different population sizes as they are identical)

- With an increasing ratio of recombination, the result quality improves–we see significant improvements over random sampling and high mutation ratio; best results are obtained with high recombination and low mutation ratio (about (80%,20%)).

4.3 TYPE-B2

As a Type-B2 problem, we study the cost distributions of the symmetric TSP where instances are given only by the coordinates of the cities. The TSPLIB collection of instances for the symmetric TSP serves as a widely accepted standard benchmark library in this field [Rei91]. In Figure 7 the cost distribution of a problem with 52 cities, obtained from 10^6 uniformly sampled tours, is depicted. The cost distribution shows the expected features: Almost all solutions are concentrated—even in the upper half of the total cost range. Moreover, they are concentrated in a very small interval. The optimal tour is known to be of length 7542. All sampled tours are longer than 21966 and shorter than 35898. Consequently, neither when randomly selecting tours for a initial population nor when adding randomly chosen tours during the optimization a tour shorter than 21966 is likely to be chosen. The best sampled tour is more than twice the length of the one found by a simple greedy algorithm (9535).

The TSP is know to be a difficult problem for evolutionary algorithms. Evolutionary algorithms when applied to this problem require special, sophisticated extensions in order to achieve competitive results (see e.g., [MW92]).

Figure 8 shows the results obtained with our implementation. We experimented with different recombination strategies found in the literature; while differing in result quality, the overall trends as depicted in the graph could be observed with all implementations. Mutation was implemented as 2- or 3-swaps.

- Both initialization and size of the population are virtually irrelevant, i.e. all random tours are significantly suboptimal and any population of nontrivial size, say, 100 or greater will lead to very similar results.

- Increasing the ratio of individuals generated by recombination leads to better result performance though results are suboptimal on average;

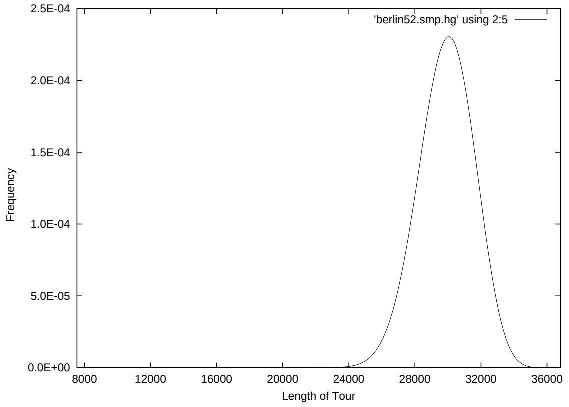

Figure 7: *Traveling Salesman Problem: Cost distribution for problem* berlin52 *of TSPLIB (optimum at 7542)*

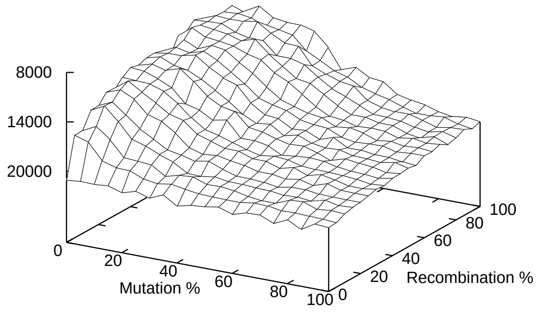

Figure 8: *Traveling Salesman Problem: Quality of optimization result as function of mutation and recombination ratio*

- Due to the low concentration of (good) neighbors, mutation can easily decrease the number of good or prospective individuals; as opposed to the other two problem types, the range of good results is smaller, around a low ratio of mutation only

- Restarts have practically no influence on the result quality.

Interestingly, we found the same trends with different recombination techniques.

5 SUMMARY

Based on the observation that a cost distribution of an optimization problem is characteristic for the problem [WGL00b], we studied its effects and implications on an optimization with evolutionary techniques. Cost distributions come in three major types of shape: a strong concentration of costs similar to the optimum (1); or else the bulk of solutions has costs either far (2) or very far (3) from the optimum.

Our analysis shows which algorithmic principles of evolutionary search are positively and which are negatively influenced by a particular shape of the cost distribution. Summing these partial influences up, we gave experimental evidence that cost distributions indicate whether a problem is (1) too easy for an evolutionary approach, i.e., evolutionary search is an overkill and simpler algorithms perform just as well; (2) of a difficulty which evolutionary techniques are typically well suited to tackle; or (3) a hard problem, where the standard repertoire of evolutionary implementation techniques achieve only mediocre performance.

Unlike previous work in this field we deliberately avoided the notion of landscape, because it is not intrinsic to the problem but artificially imposed on the space, intently or not, to allow the use of navigation algorithms. In contrast, cost distributions are entirely inherent to the problem, and independent of the optimization algorithm applied. Furthermore, we observed that cost distributions could predict the behavior of evolutionary algorithms, which do introduce and utilize a space topology. It appears that cost distributions are influential to the definition of landscapes, as the difficulty of shaping a certain landscape depends also on the number of available solutions of certain costs. For example, a landscape which is favorable for Hill Climbing optimization is significantly easier to define in case of a type-A distribution than is for a type-B.

Our analysis provides an indicator whether a given problem is difficult enough to be tackled with evolutionary algorithms; and which component of an evolutionary search technique to modify and tune, in case the results are not satisfying.

References

[Bäc96] T. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, Oxford, 1996.

[FJMO95] M. L. Fredman, D. S. Johnson, L. A. McGeoch, and G. Ostheimer. Data Structures for Traveling Salesmen. *Journal of Algorithms*, 18(3):432–479, 1995.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman and Co., New York, 1979.

[Gol89] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA, USA, 1989.

[Kau93] S. A. Kauffman. *The Origins of Order*. Oxford University Press, New York, Oxford, 1993.

[KKLO86] N. K. Karmarkar, R. M. Karp, G. S. Lueker, and A. M. Odlyzko. Probabilistic Analysis of Optimum Partitioning. *Journal of Applied Probability*, 23:626–645, 1986.

[Mit96] M. Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA, 1996.

[MT90] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley and Sons, New York, USA, 1990.

[MW92] K. Mathias and D. Whitley. Genetic Operators, the Fitness Landscape and the Traveling Salesman Problem. In *Parallel Problem Solving from Nature*, pages 219–228. Elsevier Science Publishers, 1992.

[Rei91] G. Reinelt. TSPLIB—A Traveling Salesman Problem Library. *ORSA Journal on Computing*, 3(4):376–384, 1991.

[Waa99] F. Waas. Cost Distributions in Symmetric Euclidean Traveling Salesman Problems—A Supplement to TSPLIB. Technical Report INS-R9911, CWI, Amsterdam, The Netherlands, September 1999.

[WGL00a] F. Waas and C. A. Galindo-Legaria. Counting, Enumerating and Sampling of Execution Plans in a Cost-Based Query Optimizer. In *Proc. of the ACM SIGMOD Int'l. Conf. on Management of Data*, pages 499–509, Dallas, TX, USA, May 2000.

[WGL00b] F. Waas and C. A. Galindo-Legaria. The Effect of Cost Distributions on Genetic Algorithms. Technical Report INS-R0003, CWI, Amsterdam, The Netherlands, January 2000.

[WM97] D. H. Wolpert and W. G. Macready. No Free Lunch Theorems for Optimization. *IEEE Trans. on Evolutionary Computing*, 1(1):67–82, April 1997.

[ZT99] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Trans. on Evolutionary Computing*, pages 257–271, November 1999.