Centrum voor Wiskunde en Informatica
REPORT *RAPPORT*

The Effect of Cost Distributions on Evolutionary Optimization Algorithms

F. Waas, C.A. Galindo-Legaria

Information Systems (INS)

INS-R0003 January 31, 2000

Report INS-R0003
ISSN 1386-3681

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

The Effect of Cost Distributions on Evolutionary Optimization Algorithms

Florian Waas
flw@cwi.nl

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

César Galindo-Legaria
cesarg@microsoft.com

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
U.S.A.

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, only little is known as to what criteria determine the particular difficulty of a problem.

In this paper, we address this question from an 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. We scrutinize the influence of cost distributions on the single algorithmic components of evolutionary computing.

Our analysis helps identifying (1) problems where evolutionary algorithms are overhead, (2) problems where evolutionary algorithms are highly suitable optimization algorithms, as well as (3) problems that pose difficulties for evolutionary techniques.

1991 ACM Computing Classification System: G.2.1 Combinatorics, G.3 Probability and Statistics
Keywords and Phrases: Evolutionary Algorithms
Note: Funded by HPCN/IMPACT project.

1. Introduction

Evolutionary algorithms have been shown to be successful for a variety of optimization problems like timetabling and scheduling, but there is relatively little information on what makes a problem amenable to be solved by these algorithms.

In this paper we undertake an analysis of the search space properties and explore how *cost distributions* affect the behavior of evolutionary optimization. Our approach differs fundamentally from previous work as we do not focus on space *topology*, or *landscapes*, rather, we take a step back and consider the space of solutions with *no neighborhood structure*. In this more basic view, the space is simply the set of feasible solutions, along with their associated cost.

Though landscapes do provide valuable information about how a particular algorithm approaches the problem, however, it is important to understand that any such topology is not intrinsic in the solution space of optimization problems; it is imposed artificially through a arbitrary notion of adjacency, or proximity between solutions. For example, terms like local minimum are understood with respect to a landscape, which is defined by the neighborhood structure, or connections that can be used to traverse the space. The specific adjacency however is solely defined by the implementor of the algorithm.

In contrast, the cost distribution of a search space provides basic statistical information such as the average cost of a solution, but also information whether there are concentrations of good and bad solutions.

Through a large number of experiments, we have seen that cost distributions appear to be characteristic for different optimization problems: Different instances of a given problem, say Traveling Salesman (TSP), exhibit a similar cost distribution; but instances of different problems, say TSP and Knapsack, show a distinctly different shape. Furthermore, the variation within a problem is limited and gradual.

In this paper we present a broad classification of cost distributions, based on our experimental observations, and study their effects on evolutionary algorithms. For a comprehensive analysis, we break the evolutionary framework up, scrutinize the building blocks first in isolation, and assess the resulting compound algorithm afterwards. Our model helps explain results that have been reported in the literature on the behavior of evolutionary algorithms.

The cost distributions we examined clearly indicate that some problems are "very easy", in that many good solutions exist in the space—even for problems known to be NP-complete in the worst case—, and evolutionary techniques appear overkill. Other problems appear "very hard" requiring special problem specific tuning in order to achieve more than only mediocre results. For problems in between, evolutionary algorithms turn out to be highly suitable optimization algorithms.

Our analysis can be transferred easily to assess whether evolutionary algorithms are well-suited to tackle a given problem. In case of a negative result, the analysis of components helps identify weak spots which can then be improved.

The paper is organized as follows. In Section 2 we go into more detail on space topology, and how it differs from cost distributions. In Section 3 we examine the different components of evolutionary optimization, and how they are affected by the cost distribution in the space. Section 4 reports our experimental results on three problems, Partition, Knapsack, and Traveling Salesman. Section 5 presents a summary and conclusions.

2. Parameters of Search Space

Ever since the introduction of blind search algorithms, relentless 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 motivate our choice to use cost distribution rather than topological models.

2.1 Landscape Models

Usually, the terms *topology* and *landscape* are used to describe certain properties and relations among solutions. Though a powerful tool for an interpretation of

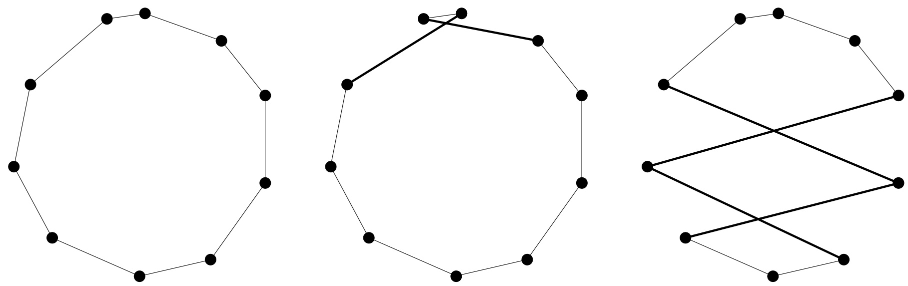

Figure 1: Alternative tours for a Traveling Salesman Problem; (a) optimal tour, (b) tour neighbored to optimal tour under N1, (b) under N2

certain effects occurring with some optimization algorithms (see e.g. [Kau93]), these structures are not intrinsic to the problem and completely different topologies can be defined easily, as the following example illustrates.

Consider for example the Traveling Salesman Problem where the shortest tour via a number of cities is sought. We 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 1 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 conversly.

Both N1 and N2 define a topology, thus a landscape, on the search space. One appears relatively smooth (N1), the other rugged (N2) although they are defined on the same problem. Neither of the two landscapes is intrinsic or natural to the problem and a variety of further neighborhood relations has been described in literature, most notably 2-swap and 3-swap. For a survey on this issue see for instance [FJMO95].

2.2 Cost Distributions

A cost distribution captures the frequencies of cost values—i.e. values of the objective function—in the complete search space. The term cost is preferable to 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.

Cost distributions are independent of the algorithm used to tackle the problem and are an invariable 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.

Cost distributions are best extracted by uniform random sampling or, if uniform generation of solutions is hard, by quasi uniform techniques like random

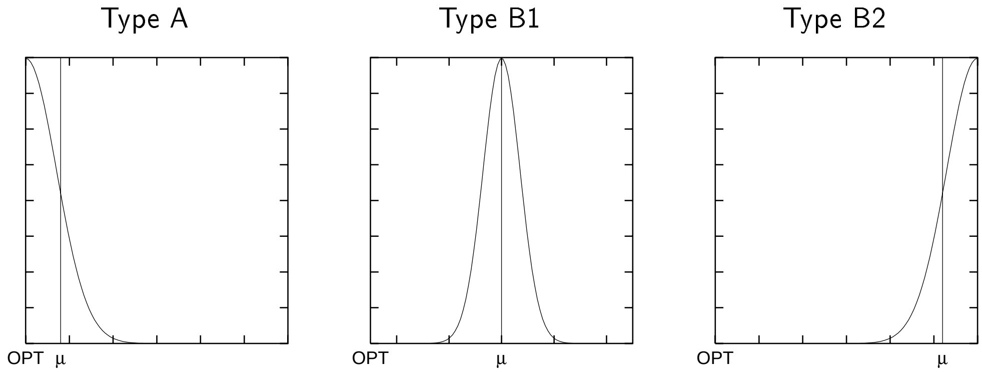

Figure 2: Basic types of distributions.

walks etc.

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—without loss of generality we assume a minimization problem—and the bulk of solutions. The question central to our further considerations is therefore:

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

In large test series with different optimization problems, we observed two basic types A and B of cost distributions, the second of which comes as two subtypes B1 and B2. In Figure 2, these distributions are shown qualitatively.

In problems with type-A distribution, the bulk of solutions 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, the cost-wise proximity does not imply any neighborhood or topology.

In problems with type-B distribution, the bulk of solutions is of distinctly different cost than the optimum. We can distinguish the subtypes 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.

Our classification distinguishes three major types of distributions, but certainly not all problems can be assigned exactly to on type, but may appear to be in between two types. However, this puts no limitation to our analysis as the results interpolate smoothly (see next section).

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, Eib96] are capable of simulating other algorithms that are commonly not considered evolutionary, like Random Sampling or Simulated Annealing [Bäc96]. In our analysis, we first sketch the different generic components of evolutionary algorithms, then scrutinize the impact of cost distributions on of 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.

Figure 3 shows an outline of a evolutionary algorithm in pseudo code (cf. e.g. [Eib96]). 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.

Let us now scrutinize the single components and how they are influenced by the different kinds of cost distributions. We will corroborate the results of our analysis with different optimization problems in the next section.

Initialization. The influence of the cost distribution on the initial phase is significant as *initializing* means literally *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 chances 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 but distinctly suboptimal 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. During this phase a *recombination* or *crossover* operator is applied to sets of individuals. It implements the actual evolutionary mechanism that mates two (or more [ERR94]) individuals and derives a new one. Strictly speaking, the result is a *random* solution consisting 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. Sophisticated algorithms are necessary to avoid a fall back to the bulk of solutions in case of a type-B2 distribution.

```
proc EvolutionaryAlgorithm
    t = 0
    P_t = ∅
    Init(P_t)
    repeat
        P'_t = SelectParents(P_t)
        Recombination(P'_t)
        Mutation(P'_t)
        P_{t+1} = Selection(P_t ∪ P'_t)
        t = t + 1
    until(done)
end
```

Figure 3: Outline of evolutionary algorithm.

Mutation. The role of mutation is disputed. It is an obvious element of natural evolution. However, it is not clear whether it is vital for evolutionary optimization techniques. For instance, Kosza suggested a mutation rate of zero [Koz91].

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 improvement by mutation is very small and mutations is only 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 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. [Rud92], [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 distributions influence on the initialization (see above).

In case of a type-A distribution, the impact of re-runs may greatly improve the results, whereas in a type-B2 scenario, re-starts do not make much of a difference.

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. However, problems of this class turn usually out to be *too* simple, rendering evolutionary algorithms an overkill. Especially hill climbing algorithms achieve a much better performance, i.e., results of comparable quality but in significantly shorter running time.

	Type-A	Type-B1	Type-B2
Initialization	$\oplus\oplus$	$\odot$	$\odot$
Recombination	$\ominus$	$\oplus$	$\ominus$
Mutation	$\oplus\oplus$	$\ominus$	$\ominus$
Restarts	$\oplus\oplus$	$\odot$	$\odot$

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

Table 1: Influence of cost distribution on components of evolutionary algorithms

Type-B1 is slightly positive influenced; this is the kind of problem evolutionary algorithms are highly suitable to optimize.

Type-B2 is the most difficult as the cost distribution has mainly negative influence. Problems of this class appear to be difficult for evolutionary algorithms to optimize.

As we pointed out before, a large number of parameters determines success and failure of evolutionary search algorithms, and negative influences of the cost distribution may be leveled—to a certain degree—by more sophisticated design of recombination operators etc. However, cost distributions indicate where advantages as well as problems are to be expected.

We also stressed that the basic framework does not restrict the choice of arbitrary crossover operators, or mutation rates. Even simulation of other optimization algorithms are possible, however, the further we swerve from the simplest of evolutionary algorithms the more likely it is that the overhead induced by the evolutionary framework becomes more and more a hindrance rather than a performance improvement and other, non-evolutionary algorithms may perform better, i.e., they find results of similar quality within shorter running time.

4. Examples

To corroborate our previous analysis we scrutinize a couple of classic, well-understood NP-complete optimization problems. We have to be aware, that the given classification is not defined in an exact way. Problems may have a cost distribution that cannot be distinctly assigned to one of the categories but appear to be in between two categories. However, in our experience, it is not useful to devise a finer system for the classification—e.g. based on the statistically characteristic values like mean, deviation, etc.—without taking further problem specific properties as well as implementation details of the search algorithm into account. On the other hand, the classification with only three categories proved surprisingly well-suited and to be of sufficient generality.

4.1 Type-A

A typical representative of this class is the number partitioning problem, where a set S of numbers is to be partitioned into 2 subsets S_1 and S_2 such that

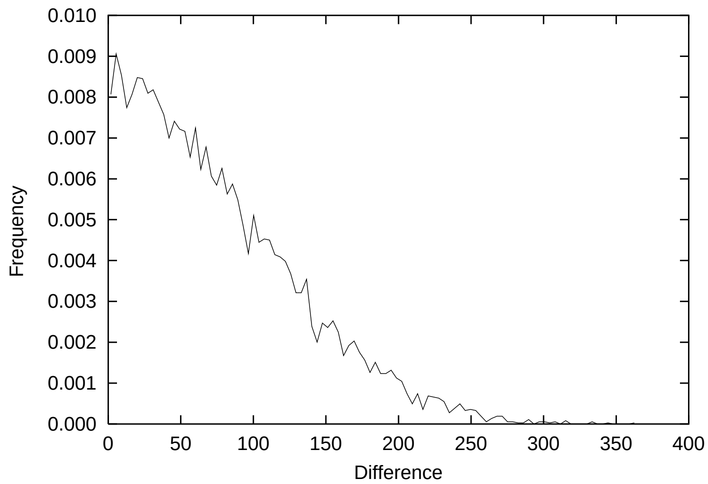

Figure 4: Cost distribution of number partitioning problem of size $|S| = 100$; obtained from a sample of size 10000

$S_1 \cup S_2 = S$, and the difference of the total sums

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

is minimal.

In Figure 4, the cost distribution of an instance with 100 elements. Optimal and near-optimal costs appear with the highest possible frequency. The distribution clearly is of the type-A kind. Even pure random sampling algorithms are very likely to find good solutions, 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 very well 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 100 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.

Problems like number partitioning, which are known to be easy to optimize may appear unrealistic and synthetic. This raises the question whether there are any practical examples that display a type-A distribution too?

In fact, some very practical real-world problems belong to this class of "easy" problems, most notably, query optimization in relational database systems. This problem ranks among the most frequently tackled NP-complete optimization problems due to the wide-spread use of database systems. Each single database query, e.g. given in a query language like SQL, poses a NP-hard optimization problem to the query optimizer.

Cost distributions found in query optimization strongly resemble exponential distributions of the type-A kind. Because of its practical relevance, this prob-

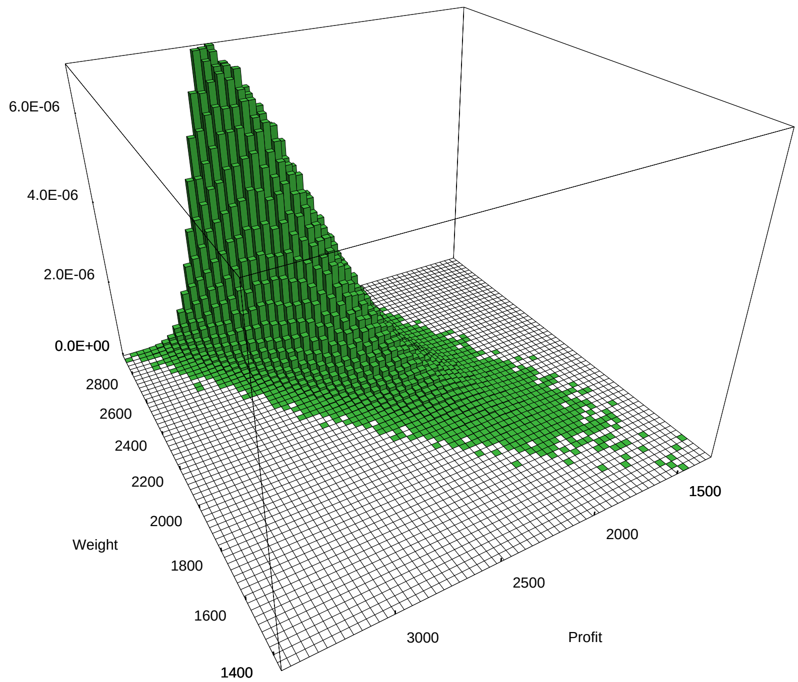

Figure 5: Cost distribution of multi-objective 0/1 Knapsack Problem, obtained from a sample of 10^6. Region of optimal solutions in the foreground.

lem has been tackled with many kinds of optimization algorithms including Genetic Algorithms and Genetic Programming. However, all three major studies published in this field [BFI91, SS96, SMK97], give experimental evidence that evolutionary search algorithms cannot outperform hill climbing techniques when applied to this problem (see also [Ste96]).

4.2 Type-B1

The class of type-B1 distributions comprises a large variety of scheduling, timetabling and assignment problems. We chose the *Knapsack Problem* as one of the representatives as its structure is easily accessible and can be well illustrated due to its 2-dimensional nature.

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, MT87]). The problem owes its name to the analogy of packing a knapsack. The variant we defined is referred to as *single-objective*, as there is only one optimization goal, the maximizing of the profit. Branch and bound algorithms as well as dynamic programming are usually the algorithms of choice [MT87, Pis95].

Let us now consider the *multi-objective* variant of the problem, where not only the profit is to be maximized but at the same time, the weight is to be minimized. As opposed to the single-objective version, there is not only one

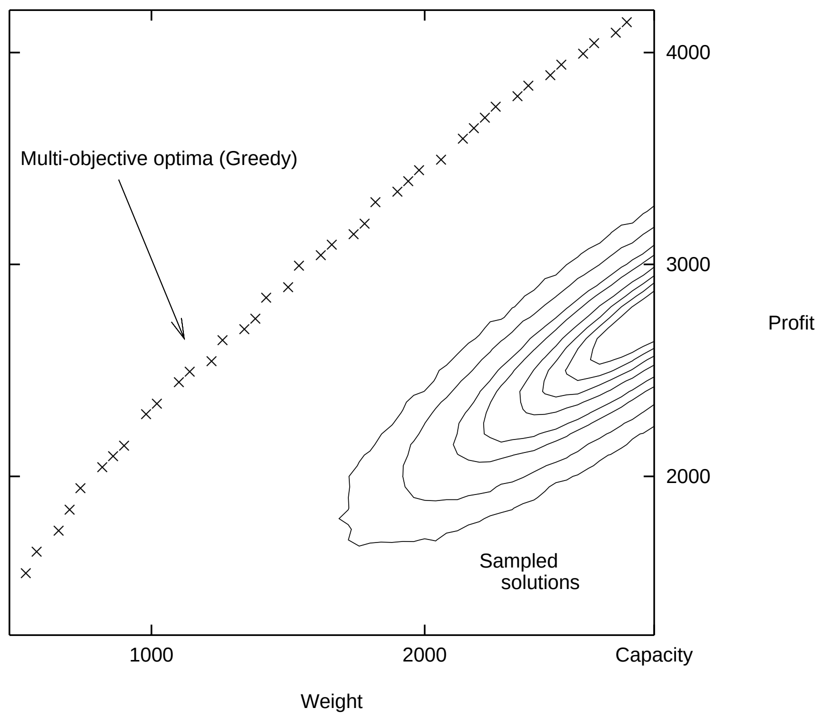

Figure 6: Cost distribution of multi-objective 0/1 Knapsack Problem (cf. Fig. 5), including best multi-objective solutions found with greedy algorithm.

single optimum but rather one optimum for each different total weight. Genetic algorithms are known to perform very well in the multi-objective case.

In Figure 5, the cost distribution of an instance consisting of 100 items is shown. The distribution was obtained by a sample of 20000 packings generated with uniform probability. 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 resulting distribution is characterized by a marginal distribution along each single weight configuration that strongly resembles normal distributions. This does not come as a surprise as both weight and profit of a particular packing of a knapsack are sums of random numbers. The shapes of the distribution appear very stable and insensitive to the problem's parameters. We conducted extensive experiments with large varieties of parameter combinations, resulting in distributions with very similar shape, the extent may differ though.

In order to give an indication of whether the distribution is of type B1 or B2, it is necessary to determine the distance of the bulk of solutions from the optima. To assess this distance, we used a greedy algorithm which provides good approximations of the optima.

In Figure 6, the distribution is shown as a contour plot. The isolines connect solutions with equal frequency. The theoretic region of optima is in direction of the left upper half, stretching from the left lower corner to the right upper corner. The optimization results found by the greedy algorithm form a lower bound of this region (see Fig. 6). The actual optima are in the close vicinity of those solutions.

The effect of this distribution on genetic search is twofold (see above): The sampling of an initial population does not contain high quality solutions. Also, the random sampling component in form of cross over and mutation is limited—the probability to sample a near optimal solution practically zero. On the other hand, the optima are not too far away from the bulk of solutions. Specifically, for the multi-objective variant, genetic algorithms are known as a suitable and very successful optimization technique.

4.3 Type-B2

Given the previous analysis, we should expect a cost distribution of a difficult problem (a) to show a strong concentration of the bulk of solutions and (b) the optimum to be far off the bulk.

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]. Figure 7 the problem 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). To check for the generality of this observation, we conducted this experiment for *all* symmetric Euclidean TSP instances given in the TSPLIB problem library, including maps of several countries, plans for drilling problems, as well as random instances [Rei91]. The shapes of the distributions observed show distinctly the features as detailed above. With increasing size of the problem, the discrepancy between the length of the tour found with a greedy algorithm $l_{min}^{(G)}$ and the best tour sampled $l_{min}^{(S)}$ compounds, i.e., $l_{min}^{(G)}$ increasingly moves to the left, $l_{min}^{(S)}$ to the right; the bulk of the distribution becomes narrower with respect to the cost range. For an in-depth analysis of the cost distributions of the TSP, including an analytical model and the explanation for the asymmetry of the total range, we refer the interested reader to [Waa99].

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]).

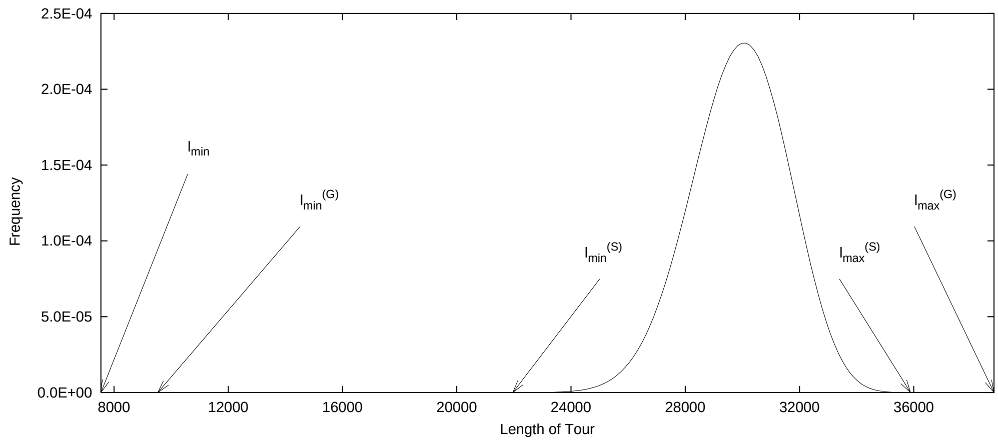

Figure 7: Cost distribution obtained from uniform sample of size 10^6. l_{min} denotes the global optimum, $l_{min}^{(G)}$ and $l_{max}^{(G)}$ shortest and longest tour found by greedy algorithm, $l_{min}^{(S)}$ and $l_{max}^{(S)}$ shortest and longest tour found by sampling.

5. Summary

Based on the observation that a cost distribution of an optimization problem is characteristic for the problem [WGL00], we studied its effects and implications on an optimization with evolutionary techniques. Cost distributions come in three major types of shape: (1) a strong concentration of costs similar to the optimum, and two variants where the bulk of solutions has costs (2) far or (3) very far off the optimum, respectively.

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 if not better, (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.

Future Work. Our findings suggest several directions for further research. For a better understanding of cost distributions an analysis of the problem specific parameters underlying is inevitable. Based on such an analysis, we pursue a classification of NP-complete optimization problems with respect to the applicability of certain optimization algorithms.

Taking this method of analysis to other optimization algorithms is an exciting challenge. An analysis of randomized algorithms like Simulated Annealing and Iterative Local Search is currently underway.

References

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

[BFI91] K. Bennett, M. C. Ferris, and Y. E. Ioannidis. A Genetic Algorithm for Database Query Optimization. In *Proc. Conf. on Genetic Programming*, pages 400–407, San Diego, CA, USA, July 1991.

[Eib96] A. E. Eiben. Evolutionary Exploration of Search Spaces. In Z. Ras and M. Michalewicz, editors, *Foundations of Intelligent Systems*, number 1079 in Lecture Notes in Computer Science. Springer-Verlag, 1996.

[ERR94] A. E. Eiben, P.-E. Raué, and Zs. Ruttkay. Genetic Algorithms with Multi-parent Recombination. In *Parallel Problem Solving from Nature*, number 866 in Lecture Notes in Computer Science, pages 78–87. Springer-Verlag, 1994.

[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.

[Koz91] J. R. Koza. *Genetic Programming.* MIT Press, Cambridge, MA, USA, 1991.

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

[MT87] S. Martello and P. Toth. Algorithms for Knapsack Problems. *Annals of Discrete Mathematics*, 31, 1987.

[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, Amsterdam, The Netherlands, 1992. Elsevier Science Publishers.

[Pis95] D. Pisinger. *Algorithms for Knapsack Problems.* PhD thesis, University of Copenhagen, Copenhagen, Denmark, 1995.

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

[Rud92] G. Rudolph. Parallel Approaches to Stochastic Global Optimization. In *Proc. of the European Workshop on Parallel Computing*, pages 236–247, Barcelona, Spain, March 1992.

[SMK97] M. Steinbrunn, G. Moerkotte, and A. Kemper. Heuristic and Randomized Optimization for the Join Ordering Problem. *The VLDB Journal*, 6(3):191–208, August 1997.

[SS96] M. Stillger and M. Spiliopoulou. Genetic Programming in Database Query Optimization. In *Proc. Conf. on Genetic Programming*, pages 388–393, Stanford, CA, USA, July 1996.

[Ste96] M. Steinbrunn. *Heuristic and Randomised Optimisation Techniques in Object-Oriented Database.* DISDBIS. infix, Sankt Augustin, Germany, 1996.

[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.

[WGL00] F. Waas and C. A. Galindo-Legaria. The Effect of Cost Distributions on Genetic Algorithms. Technical report, January 2000. Submitted for publication.

[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.