Proceedings of the 5th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Madrid, Spain, February 15-17, 2006 (pp123-126)

Combinatorial Complexity: Are We on the Right Way?

ROBIN GREMLICH, ANDREAS HAMFELT, HÉCTOR DE PEREDA, AND VLADISLAV VALKOVSKY
Department of Information Science
Uppsala University
Box 256, 751 05 Uppsala
SWEDEN

Abstract: - We present a hypothetical approach to support existing methods in dealing with combinatorial complexity and describe the application of said approach to some typical combinatorial problems and techniques.

Key-Words: - Combinatorial complexity, NP-complete problems, Traveling Salesman Problem

> "Glory to God who created difficult
> unnecessary and unnecessary difficult."
> Grigory Skovoroda (1722-1794)

1 Introduction

Analyzing the history of Artificial Intelligence (AI) shows that current achievements in the field are less impressive than was expected during the "romantic period", when the AI sphere was created. There are a several reasons for this state, which are out of the scope of this paper, but one of the most important is the combinatorial complexity of AI problems. Because of combinatorial complexity currently we only have decisions for relatively small dimensions for many AI problems and far from the dimensions of real large-scale problems needed for industrial applications. Many AI problems are from the class of so called NP-complete problems [1] and till now it is an open question whether or not NP = P? This question is one of the seven problems, which were declared year 2000 as Millennium Prize Problems by The Clay Mathematics Institute of Cambridge, Massachusetts [2].

To overcome combinatorial complexity there have been proposed a multitude of approaches: operational research methods, constraint programming techniques, randomized algorithms, genetic algorithms, neural networks, etc.

Despite of great achievements in these fields till now decisions of large-scale combinatorial problems remain problematic. At the same time a lot of practical applications (DNA-sequencing, bio-informatics, data mining, statistical physics, etc.) currently are needing decision of combinatorial problems for huge (in limit infinite for statistical physics) number of dimensions.

So, the question is: what would be the way to support existing methods in dealing with combinatorial complexity?

2 What is a possible way?

Let us look at the paradigm of statistical mechanics. The ideal gas law formula is:

$$P \cdot V = R \cdot T$$

Here P is pressure, V is the volume of the gas, R is gas constant and T is temperature.

This law is precise despite the fact that the objects (molecules) are stochastic.

The precision of the law is due to the tremendous number of objects in the analyzed system, which implies that the system exhibits a regular behavior in average.

So the paradigm of statistical mechanics is:

Stochastic → Deterministic

Imagine deciding large-scale combinatorial problems inversely to the statistical mechanics paradigm. It means that we will treat deterministic data just the other way around - as stochastic data and will try to predict the deterministic combinatorial decision by statistical categories (e.g. by the prediction of distribution density for the elements participation in the optimal solution). Distribution density does not depend on permutation of sample elements and therefore it would be possible to cut down drastically the computational complexity of combinatorial problems. So, the proposed paradigm is:

Deterministic → Stochastic

But what are we paying for decreasing complexity by such a paradigm? We are losing the ordering of our initial data and as a result we are losing the ordering of data in the decision, because in this case our decision will also be described in statistical categories. However such knowledge about the distribution of the optimal decision could help us to

reduce the search space of the problem by dividing initial data into two groups: a group of initial data which are relevant for the optimal decision and a group of data which are irrelevant for the optimal decision. More over, knowledge about the distribution density for elements of the optimal decision could support us in creating algorithms for finding an optimal (suboptimal) combinatorial problem solution.

3 Applying the paradigm to TSP

The proposed paradigm was applied for deciding the Traveling Salesman Problem (TSP) [3]. The reason for using TSP as benchmark is its NP completeness and the fact that all NP complete problems could be transformed one to another in polynomial time [1].

The TSP is stated as follows [4]: Given a finite number of "cities" along with the cost of travel between each pair of them, find the cheapest way of visiting all cities and returning to the starting point.

Although it seems to be quite simple, solving it is very computationally expensive. As no polynomial time algorithm has been discovered, the only way to obtain the best solution is by calculating all possible routes. If we have a map with N cities, the number of possible routes will be N!. For example solving the problem for 30 cities, using a computer with 10^9 adds per second, would take over $8 \cdot 10^{15}$ years.

As was shown in [3] for the general (asymmetric) case of TSP, it is possible to predict for a given TSP weight matrix (matrix of all distances between cities) the distribution density for the elements of weight matrix which will participate in optimal decision.

For light left tail TSP weight matrixes distributions it will be Rayleigh distribution:

$$p(x) = \frac{x}{\sigma^2} \cdot e^{-\frac{x^2}{2 \cdot \sigma^2}}$$

For heavy left tail TSP weight matrixes distributions the optimal decision distribution will be γ-distribution:

$$p(x) = \frac{1}{\Gamma(\alpha+1) \cdot \beta^{\alpha+1}} \cdot x^{\alpha} \cdot e^{-\frac{x}{\beta}} \qquad \alpha = 0.5$$

Methods for finding parameters of these distributions by a given asymmetric TSP weight matrix, calculating a prediction of the optimal tour length, its variance, and asymptotic behavior analysis for the predicted TSP optimal weight decision and its variance are also presented in [3].

The result of this research provides an opportunity to reduce drastically the search space for solving asymmetric TSPs. For example, for the dimension of the problem n = 1000 we can delete nearly 99% of the elements of the weight matrix as irrelevant for the optimal decision. For the dimension n = 10000 nearly 99.9 % of the elements can be deleted, etc. After such a pruning according to predicted optimal decision distribution we can determine the probabilities for the remaining elements of the weight matrix to be included in the optimal decision to support finding an optimal TSP solution.

4 How could these results be used?

Now let us look shortly on possible usage of these results.

4.1 Open TSP problems

In [4] the following open theoretical TSP problems were formulated.

Problem 13

Establish that:

$$\frac{T(U_1,...,U_n) - ET(U_1,...,U_n)}{(VarT(U_1,...,U_n))^{1/2}} \to N(0,1)$$

$T(U_1,...,U_n)$ *– The length of the shortest path.*

$U_1,...,U_n$ *– Uniformly distributed random variables on the unit square.*

$VarT(U_1,...,U_n)$ *– Variance for the length of the shortest path.*

$N(0,1)$ *– Normal distribution with mathematical expectation 0 and standard deviation 1.*

As it was shown in [3] the length of the shortest path S can be represented as the sum of n independent probabilistic values of Rayleigh distribution with known mathematical expectation MX and variance DX. By the central limit theorem it follows:

$$\frac{S - n \cdot MX}{\sqrt{n} \cdot \sqrt{DX}} \to N(0,1)$$

Problem 16

Establish that the variance of the TSP converges, i.e. show that in dimension 2 we have $VarT(U_1,...,U_n) \to C$, *where* C *is some positive constant.*

$U_1,...,U_n$ *– Uniformly distributed random variables on the unit square.*

$VarT(U_1,...,U_n)$ *– Variance for the length of the shortest path.*

It was shown in [3] that for the case of the points distributed randomly and uniformly on the unit square there is the convergence of the variance and $\lim_{n\to\infty} \sigma^2(L_E) \approx 0.1385$.

4.2 Statistical physics

Based on an analysis of experimental results, Percus et al [5] propose the hypothesis that for the case of n points, distributed randomly and uniformly in a unit square (Euclidean case), there is a limit for variance $\sigma^2(L_E)$ of the optimal tour length L_E, where $\lim_{n\to\infty} \frac{L_E}{\sqrt{n}} = \beta_E$ exists and $\beta_E = 0.7120 \pm 0.0002$.

"...the variance remains relatively constant... and in our Euclidean study σ being approximately half of 0.59 value" [5].

As it follows from Problem 16:

$$\frac{\sigma(L_E)}{0.59} = \frac{\sqrt{0.1385}}{0.59} \approx 0.63$$

In statistical physics the optimal decision of the TSP is precisely equivalent to finding the system's lowest-energy state. Therefore an opportunity to find in the general case the distribution function of the optimal TSP decision for the given distribution function of the initial state of the system could be theoretically interesting.

4.3 Genetic algorithms

In genetic algorithms one of the main problems is detecting and avoiding local minimums. In [6] by introducing the concept of Oracle, which is predicted distribution density for TSP optimal decision, an approach was proposed to detect and avoid local minimums. By Oracle it is possible to know the expected length of the optimal route and its variance. This information is used for evaluating whether we have found an appropriate solution or we have to keep on searching. Local minimums are detected by counting the number of generations that have been created without improving a best solution, which is sufficiently far from the one predicted by Oracle. When this number reaches a certain quantity, it means that we are in a local minimum and we have to act accordingly. The way of skipping a local minimum is going back in the evolution and trying to evolve in a different direction. This was accomplished by mutating a whole population enough so it does not get back to the same solution we had in the local minimum, but not so much that we loose all the evolution we had already performed.

One more problem of genetic algorithms is generating the initial population.

Starting from a sufficiently good initial population can save a lot of evolution time, as well as facilitating the genetic algorithm to deliver better quality solutions. Our task was to create an efficient and quick way to create initial population members. We were using the optimal decision distribution predicted by Oracle as a guide that can lead us to solutions looking quite similar to the optimal path.

We came up with the idea of sorting each node's neighbors by predicted Oracle distance density distribution. Then, when it comes to choose the next node in the path, we pick an unvisited neighbor with a probability that is directly proportional to its distance expected density. In this way, distances with an insignificant density will not be picked unless the rest of the neighbors have been visited already. This method reminds of the nearest-neighbor algorithm [7], but introduces a pseudo-random factor and considers the expected distance density in the final path instead of the distance itself. We call this method most-probable-neighbor algorithm. Fig.1 represents the outstanding performance of the new algorithm, which, in addition, was not as time consuming as preliminary attempts. For each TSP size from 10 to 500, we calculated the average total length of initial population members, both generated randomly and using the most-probable-neighbor algorithm. As we can see, the members of initial populations generated by this new method are much closer to the optimal route than the solutions provided by genetic algorithms before implementing this improvement, even after performing genetic algorithms' evolution through a huge number of generations. This will allow us to start the evolution process from a point where it would have taken an enormous amount of time to get to applying mutations.

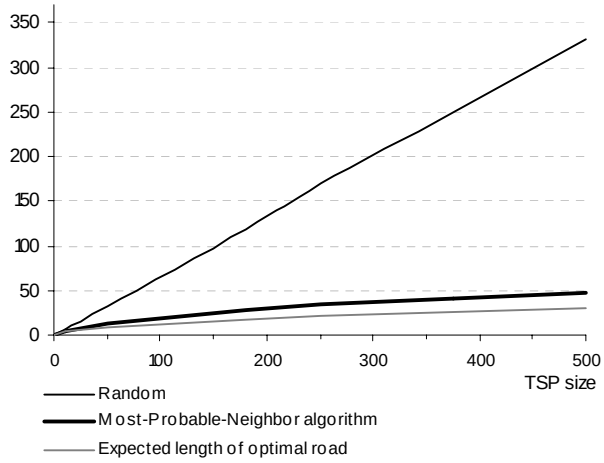

Fig.1 Average length of initial population members

By combining techniques for avoiding of local minimums and generation of initial population, the final algorithm was proposed. If we take a look at the different distribution densities we have obtained, we will be able to notice how each time they get closer to the optimal decision distribution density predicted by Oracle, as shown in Fig. 2.

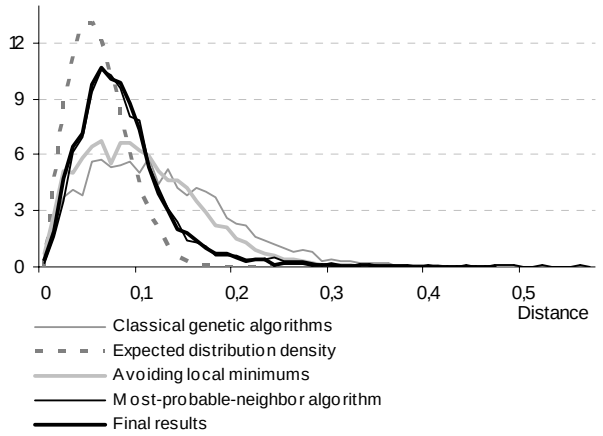

Fig.2 Distribution density

4.4 Constraint programming

Constraint programming is a very powerful and flexible technique for deciding combinatorial problems [8]. But for some combinatorial problems (and TSP in particular) target optimization function is loosely connected with decision variables, so it is not a trivial task to define proper constraints for them. This is the reason why the decision of such kind of problems by a constraint programming approach is restricted to not so big dimensions (for TSP in particular it is restricted to dimensions of a few hundreds [9]).

Using the proposed approach it would be possible not only to define proper constraints but also to filter initial data for relevant and irrelevant parts to reduce the search space of the problem.

5 Conclusion

There are different opportunities to use the proposed approach to support solving of large-scale combinatorial problems.

One way is to transform a NP-complete problem to TSP, finding a solution (optimal or suboptimal) in the TSP statement with support of the proposed paradigm and then transforming this decision back to the initial problem (it is possible to do quickly because all NP-complete problems are transformable one to another in polynomial time).

Another opportunity could be transforming the NP-complete problem to TSP, finding the distribution density for the optimal decision in the TSP statement by the proposed approach and then transforming this distribution or its parameters (mathematical expectation, variance, confidence intervals, etc.) back to the initial task in order to reduce the search space for solving the problem.

References:

[1] Papadimitriou C.H., Steiglitz K., *Combinatorial Optimization, Algorithms and Complexity*, Englewood Cliffs, NJ, Prentice Hall, 1982.

[2] http://www.claymath.org/millennium

[3] Robin Gremlich, Andreas Hamfelt, and Vladislav Valkovsky, Prediction of the Optimal Decision Distribution for the Traveling Salesman Problem, *Proceedings of IPSI International Conf.*, Sveti Stefan, Montenegro, 2004.

[4] Gutin, Punnen (eds.), *The Travelling Salesman Problem and its Variations*, Kluwer Academic Publishers, 2002.

[5] A.G. Percus, O.C. Martin, The Stochastic Traveling Salesman Problem: Finite Size Scaling and the Cavity Prediction, *Journal of Statistical Physics*, Vol. 94, Nos. 5/6, 1999, pp. 739-758.

[6] Robin Gremlich, Andreas Hamfelt, Héctor de Pereda, and Vladislav Valkovsky, Genetic Algorithms with Oracle for the Traveling Salesman Problem, *Proceedings of Enformatika International Conf.*, Vol.7, Prague, 2005, pp. 27-32.

[7] Cover T.M., Hart P.E., Nearest Neighbor Pattern Classification, *IEEE Transactions on Information Theory*, 13, 1967, pp. 21-27.

[8] Kim Marriot, Peter J. Stuckey, *Programming with Constraints*, MIT Press, 1988.

[9] Michela Milano, Willem J. van Hoeve, Reduced Cost-Based Ranking for Generating Promising Subproblems, *Proceedings CP'2002*.