Napoli. November 2005

Contents

List of Figures

List of Tables

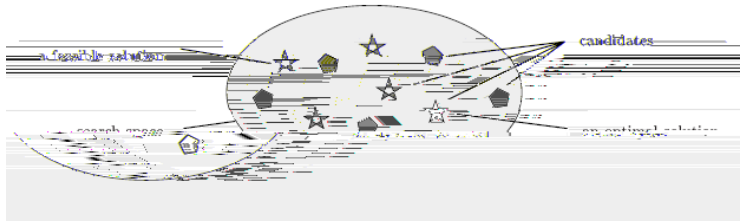
Introduction

performance and is functionally related to parameters of the system. An important and often the most difficult step in an optimization process is to define an appropriate objective function for the problem at hand. There may exist multiple objectives at the same time, and the relative weights of each of these corresponding goals should be considered as well(Deb, 1999).

1.2 Combinatorial optimization problems

1.2.1 Some definitions

As mentioned before, many usual computational problems amount to searching


candidates

> Given a graph *G*, is there a path of length shorter than *l* that visits each vertex of G exactly onc3(e)-?

Th434(des)-1(cri)1(ption)-433(of)-434(a)-433(prob)1(lem)-435(i)1(s)-435(n)1(ot)-434(ge)-1(n)1(e)-1(r
TSP is a problem and the TSP together with a given graph (thato-288(rep)1(re)-1(sen)28(ts)-288(th)
in the remainder of this work the distinction won't be done explicitly unless some ambiguities are possible.

1.2.2lasses ofoblems

A complete description of the di□erent classes of combinatorial optimization problems, according to their complexity, candsolvro,□11.95ritit1(r)1ou3(,)-50C.s,

is considered to be computationally intractable in the sense that an algorithm able to solve them requires an exponential amount of time because, in the worst case, such algorithm would need to enumerate all possible candidates from the solutions space. This leads to the fact that only very small instances of such problems can be solved within a reasonable amount of time and large ones are computationally prohibitive.

characteristic of hill-climbing methods is that they begin with an initial solution (usually determined randomly or deterministically by users) and a new solution is calculated based on the current solution. The algorithms differ from each other according to the update rules used to create a solution. In general, direct optimization methods are computationally prohibitive, and they tend to work for simple unimodal functions or specialized applications. On the other hand, indirect methods require the calculation of gradients of functions and constraints. Most of the these methods do not guarantee to find the global optimal solutions, because these algorithms usually terminate when the gradient of the objective function is very close to zero, which may occur both in case of local and global solutions. The other obvious drawback of indirect methods is the calculation of gradient, which may be expensive or not well defined. In many real world combinatorial optimization problems the gradient of the objective function and constraints may not be calculated exactly because the objective function and/or

behind the method is to explore the search space of all feasible solutions by a

1.4.6 Evolutionary Programming

Evolutionary programming (EP) was developed by Lawrence Fogel in the late 1960s. Although EP techniques originally aimed at evolving artificial intelligence in the sense of developing the ability to predict changes in the environment, it is often used as an optimizer. After initializing a population of N individuals and generating N children by mutation, N survivors are selected from the population of parent and children using a probabilistic function based on fitness. As a consequence, individuals with a greater fitness have a higher chance to survive to the next generation.

1.4.7 Evolution Strategies

Evolution strategies (ES) are algorithms that mimic the principles of natural evolution as a method to solve parameter optimization problem. They were developed hildd iny t lmoste that genetic algorithms emerged in the U.S.A. during the late 1960s. Early evolution strategies were based on a population consisting of one individual and one genetic operator (mutation) only. This method called the two-member evaluation strategy. However, the most significant di□erence between simple genetic algorithms and evolution strategies is the representation of the variables. In ES, an individual was represented as a pair of floating point-valued vectors, i.e., $v = (x, \sigma)$. Here, the first vector x represents point in theion space; thecond vector σ represents

the corresponding strategy parameter ... individuals during the generation. This ...)1(.)-430(Th)1(e)-289(ind)1(ivid)1(ual)-288(

alHs tmoreβtindividualstolivef torvweryoloingm, theretcausesf TJΦdtheprematuenconvwergenceIonereent

olingmoniβzationismplemsentedtothemlatimemberevolutionstrategyTJΦdHreapoputationconsrsvingof TJΦf

Chapter 2

Classification and Regression Trees

2.1 Introduction

Classification and Regression Trees (CART) is a relatively young technique developed by a group of American scientists [] during the last 25 years. As it is for discriminant analysis, the aim of CART is to classify a group of observations or a single observation into a subset of known classes, on the basis of a particular variable. Comparing to classical parametric discriminant analysis

radical changes in decision tree: increase or decrease of tree complexity,

Figure 2.2: An example of maximum tree.

the learning sample with variable matrix X with M number of variables x_j and N observations. Let class vector Y consist of N observations with total amount of K classes. The Classification tree is built in accordance with splitting rules - the rule that performs the splitting of learning sample into smaller parts. In each step of the algorithm data have to be divided into two parts with maximum homogeneity, as shown in Figure 2.3: where t_p, t_l, t_r are, respectively, parent,

Figure 2.3: Splitting from CART

will consider only the Gini splitting rule, that is based on the following impurity function:

$$i(t) = \square$$

In order to get the best absolutely best Tree "it would be enough", in theory, to examine all possible generable trees from the learning sample in order to choose the one that minimizes the overall misclassification rate $R(T)$. The problem is that this is computationally unfeasible because of the fact that the number of generable trees grows exponentially with the complexity of the problem so, even for very small learning samples with not too many variables, such operation would take a huge amount of computation. This led to the choice to use a greedy algorithm[6] for selecting the maximum tree. Such algorithm, which starts by the root node, is described as follows:

- function greedy(Node n) *{*

  –

where $Var(Y_l), Var(Y_r)$ are calculated on the response vectors for corresponding

Figure 2.5: Removing the branch under the node 3

for Classification Trees and the η^2 index for Regression Trees, demonstrated that it is possible, by making use of the forementioned indexes, to reduce the

6. If $\tau(Y|s_k) < \tau(Y|X_{k+1})$ then compute $k = k + 1$ and go back to step 3. Otherwise, stop the procedure, being $s = s_k$ the best split.

2.4 The J-FAST project

A Java software has been developed (see[]) in order to get classification and Regression Trees by making use of the FAST algorithm. Such software o□ers interactive visualization and comparison interface in order to give a more transparent view of the whole tree building process. The J-FAST project consists of the Java program and of an HTML-based tutorial, which will be presented in the following section.

2.4.1 The J-FAST program

The J-FAST program is a Java-based segmentation software which is particu-

Figure 2.8: More than one tree on the screen for comparisons

- Split: binary split
- TreeGrower: class that builds trees
- Pruner: class that takes care of pruning
- TreeViewer: interactive interface class
- Utility: many useful function like reading data from plain text files, excel sheets, etc.

J-FAST has also some main disadvantages, which are reported as follows:

- J-FAST is a very young project. It is still buggy and a good testing period is necessary to find and fix all the bugs.

2.5. Forward Search to improve stability of Regression Trees

in order to split it and minimum terminal node size) on the robustness of the procedure. It is assumed that the chosen robust subset contains the most ho-

Chapter 3

Genetic Algorithms

Organisms whose chromosomes are arrayed in pairs are called *diploid* (see [])

3.1. Genetic Algorithms in detail

Elitist Scheme

The elitist selection scheme was proposed by De Jong. T99.8is met8od copies a certain number of the best individuals from the existing population to the next population. This enforces preserving the best structures for the problem at hand.

Rank Selection

A nonparametric procedure for selection (*Rank Selection*

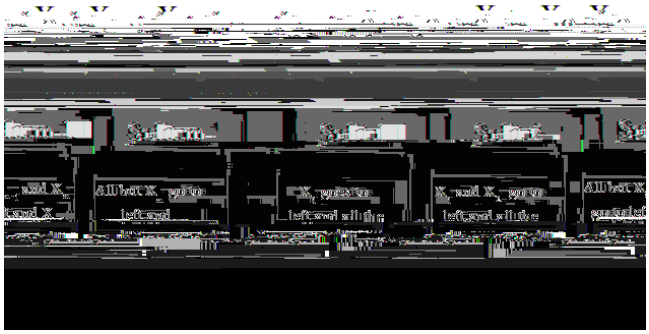


Figure 3.9: Building splits for an ordinal predictor

by such predictors grows exponentially with the number of modalities m. The number of pos

Figure 3.10: Encoding a split for a nominal predictor

- end while
- return best solution

3.2.6 Application on simulated and real datasets

solution of a problem by exchanging information via pheromone deposited on graph edges. Ant System has been applied to many combinatorial optimization problems such as the traveling salesman problem (TSP) [], and the quadratic assignment problem [] by making use of the so-called Ant Algorithms. As the attention is not on simulation of ant colonies, but in the use of artificial ant colonies as an optimization tool, Ant system have some major differences with a real (natural) one:

- artificial ants have some memory
- they are not completely blind
- they live in an environment where time is discrete

Figure 4.2 shows a possible Ant System interpretation of the situation in figure 4.1. In this case the distances between the points D and H, B and H, B and

toward H and 15 toward C (Fig. 2b). At t=1 the 30 new ants that come to B

Figure 4.3: An instance of the TSP problem

- it chooses the town to go to with a probability that is a function of the town distance and of the amount of trail present on the connecting edge;
- it is forced to make legal tours, i.e., tours in which already visited towns are disallowed until a tour is completed (this is done by a tabu list);
-

between time t and $(t + n)$. Such quantity is given by

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} & \text{i} \end{cases}$$

4.2.1 The algorithms

Given the definitions of the preceding section, the so-called ant-cycle algorithm is simply stated as follows. At time zero an initialization phase takes place during which ants are positioned on different towns and initial values $t_{ij}(0)$ for trail intensity are set on edges. Dorigo[] suggests to use, as initial trail

2. Set s= 1;

   For k = 1 to m do

   Place the starting town of the k-th ant in $tabu_k(s)$

3. Repeat until tabu list is full

   Set s=s+1;

   For k = 1 to *m* do

   Choose the town *j* to be moved by the k-th ant, not yet visited, according to the transition probability $p_{ij}^k(s)$

   Move the k-th ant to the town *j*

   Insert town *j* in $tabu_k(s)$

not only interested to the minimum tour length, but was also worried about the stagnation phenomenon that is a situation in which all the ants follow the same path and construct the same tour at each iteration. This is not a good feature for a random search algorithm because it means that no new solutions are being explored. While the parameters tuning phase of the algorithm (which is made empirically because a formal model able to describe the system's behavior for di□erent parameters settings is not available) the stagnation phenomenon indicates that the current parameter settin is not optimal. Dorigo ran di□erent simulations and came to following conclusions about Ant System algorithms:

- For Ant-deny
    1. The best parameters are: $\alpha = 1, \beta = 10$ and ρ as big as it can be (0.999).
    2. The Q parameter has shown to be not relevant
    3. The stagnation phenomenon appears only (after 200-300 iterations) for values of $\beta \geq 2$.
- For Ant-quanti2y
    1. The best parameters are $\alpha = 0.5, \beta = 20$ and ρ as big as it can be.
    2. The Q parameter has shown to be not relevant3.

is no experimental evidence in favor of that about the quality of the solutions. ACS is based on an earlier algorithm proposed by [] which was called Ant-Q. The only di□erence between Ant-Q and ACS is the definition of the term τ_0 which in Ant-Q is set to $\tau_0 = ymax_{j \quad N}$

by tentatively adding the arc to the current partial solution and by estimating the cost of a complete tour containing this arc my means of a lower bound. Such estimate is used to compute η_{ij}: the lower the estimate the more attractive the addition of a specific arc is. Such approach has an advantage and a disadvantage: the former lies in the fact that otherwise feasible moves can be discarded if

Figure 4.5: Effects of early stagn0S0(tion)]TJET Q0013150.2-462.711cm0g0G100

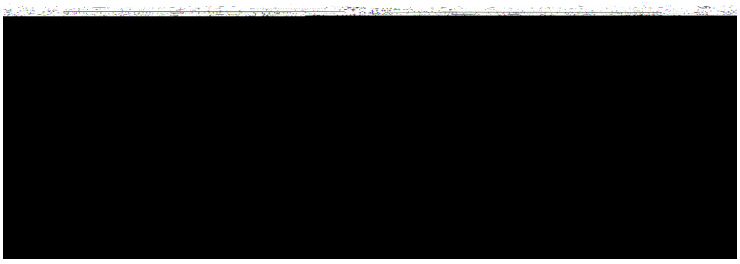
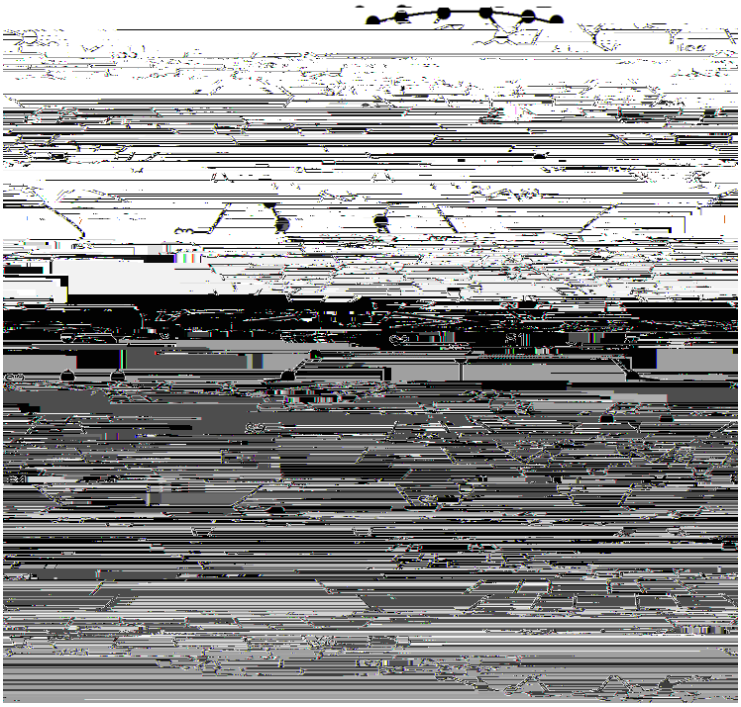


Figure 4.7: Performance of AS and its extensions

λ

having the lowest global impurity measurement among all possible generable trees. It has been shown [] that constructing the optimal tree is a NP-Complete problem. This means that it, in order to use a polynomial algorithm, it is only possible to get suboptimal trees. In other words the well-knows segmentation procedure make use of greedy heuristics in order to reach a compromise between tree quality and computational e□ort. In particular, most existing methods to build decision trees use the following heuristic, which is based on a top-down approach:

- recursively do the following until no more nodes can be split



distance between two nodes. Once the construction graph has been built and

Conclusions and

interesting.
Chapter 4.1, finally, proposes an Ant Colony Optimization based algorithm for improving the results obtained by the greedy search algorithm commonly used by CART. It also give encouraging results.