Generating True Alternatives with a Penalty Method

Ingo Althöfer
Institut für Angewandte Mathematik
Friedrich-Schiller-Universität Jena
07740 Jena - Germany
althofer@minet.uni-jena.de

Franziska Berger
Zentrum Mathematik
Technische Universität München
80290 München - Germany
berger@mathematik.tu-muenchen.de

Stefan Schwarz
Institut für Angewandte Mathematik
Friedrich-Schiller-Universität Jena
07740 Jena - Germany
delgado@minet.uni-jena.de

June 11, 2002

Abstract

In many applications k-best algorithms produce only micro mutations of the optimal solution instead of true alternatives. A penalty method gives better chances to find true alternatives. For minimization problems with sum-type objective function this approach generates alternatives whose dissimilarity to the original optimum grows monotonically with the penalty parameter.

Keywords: Multiple Choice System, k-best algorithm, micro mutation, true alternative, sum-type problem.

1 Introduction

Multiple Choice Systems are specific decision support systems: A computer generates a clear handful of candidate solutions, and a human boss makes the final choice amongst these alternatives [Alt 00a].

A natural way to get alternative solutions seems to be to use a k-best algorithm, for some appropriate number k (k=2 or 3 or 5). However, in most applications the k best solutions tend to be very similar to each other. Typically they are merely micro mutations of the best solution instead of true alternatives. In the Eppstein internet bibliography [Epp 90++] on k-shortest path problems several discussions on this can be found, for instance [NSZZ 94].

A penalty method gives better chances to find true alternatives: The best solution is computed, then certain building blocks are penalized. The best solution with respect to this modified objective function represents an alternative solution. Of course, different penalty parameters typically lead to different alternatives. For sum-type problems this approach generates alternatives whose dissimilarity to the original optimum grows monotonically with the penalty parameter. At the same time the relative loss in quality (with respect to the global optimum) is also monotonically bounded by the penalty parameter.

Section 2 contains the basic technical result. Consequences for uniform penalties are exhibited in Section 3. In Section 4 we mention some exemplary applications in OR and Computational Biology. A list of theses and three open problems in Section 5 conclude the note.

2 The Basic Technical Result

This following theorem for general sum-type problems has wide-spread applications. Its proof is very straightforward, but we did not find it elsewhere.

Let E be a finite set, and let C be a familiy of feasible subsets of E. Let $r : E \to \mathbb{R}$ and $s : E \to \mathbb{R}$ be two real-valued functions on E. For every element B in C we define $r(B)$ to be the sum of the $r(e)$, taken over all building blocks e in B. Analogously $s(B)$ is defined. For a real-valued penalty parameter ε we define

$$f_\varepsilon(B) = r(B) + \varepsilon * s(B) \tag{1}$$

for all B in C.

Here and below "$*$" is the sign for normal multiplication. $r(\ )$ is the original objective function which is to be minimized. $s(\ )$ is the penalty function, $s(e) > 0$ means that building block e is punished. $\varepsilon \geq 0$ is the penalty parameter. In case of $\varepsilon = 0$ there are no penalties at all. The minimum of f_ε gives the alternative solution for parameter ε.

Theorem: Let B_ε be some set B in C with minimum value for f_ε. Two statements of monotonicity hold for nonnegative ε.

(i) $s(B_\varepsilon)$ is weakly monotonically decreasing in ε.

(ii) $r(B_\varepsilon)$ is weakly monotonically increasing in ε.

Proof. Take two parameters $0 \leq \varepsilon_1 < \varepsilon_2$. By the optimality of B_{ε_1} and B_{ε_2} for their respective parameters we have

$$\begin{aligned} f_{\varepsilon_1}(B_{\varepsilon_1}) &\leq f_{\varepsilon_1}(B_{\varepsilon_2}) && (2) \\ f_{\varepsilon_2}(B_{\varepsilon_2}) &\leq f_{\varepsilon_2}(B_{\varepsilon_1}) && (3) \end{aligned}$$

Summing up the inequalities (2) and (3), using (1), and sorting the terms gives

$$(\varepsilon_2 - \varepsilon_1) * s(B_{\varepsilon_2}) \leq (\varepsilon_2 - \varepsilon_1) * s(B_{\varepsilon_1}) \qquad (4)$$

Division by the positive value $(\varepsilon_2 - \varepsilon_1)$ proves statement (i).

The proof of (ii) takes one more preparatory step. Multiply inequality (2) by factor ε_2, and (3) by ε_1. As the ε's are nonnegative this does not change the inequality signs. Then sum up the inequalities, use (1), and remove the identical terms. It remains

$$\varepsilon_2 * r(B_{\varepsilon_1}) + \varepsilon_1 * r(B_{\varepsilon_2}) \leq \varepsilon_2 * r(B_{\varepsilon_2}) + \varepsilon_1 * r(B_{\varepsilon_1}) \qquad (5)$$

Sorting the terms and division by $(\varepsilon_2 - \varepsilon_1)$ shows that $r(B_{\varepsilon_1}) \leq r(B_{\varepsilon_2})$. This completes the proof of the theorem.

3 Two Special Cases with Uniform Penalties

First we look at the case where building blocks are punished by multiplying their original weights $r(e)$ by a constant factor $(1 + \varepsilon)$ for some $\varepsilon > 0$.

Let B_0 be a minimum solution for the original sum-type problem with weights $r(e)$. Define s on E by

$$s(e) = \begin{cases} r(e), & \text{if } e \in B_0, \\ 0, & \text{if } e \notin B_0. \end{cases}$$

Thus we get for $\varepsilon \geq 0$

$$\begin{aligned} f_\varepsilon(B_0) &= (1 + \varepsilon) * r(B_0)\,, \\ f_\varepsilon(B) &= r(B) \qquad \text{for all } B \text{ which are disjoint to } B_0\,, \end{aligned}$$

and in general $r(B) \leq f_\varepsilon(B) \leq (1 + \varepsilon) * r(B)$ for arbitrary B in C.

Let B_ε be *a* (an arbitrary) minimum solution for f_ε. Then statement (ii) of the theorem gives that

(A1) $r(B_\varepsilon)$ is weakly monotonically increasing in ε.

From statement (i) it follows that

(A2) $r(B_\varepsilon \cap B_0)$ is weakly monotonically decreasing in ε.

Taking the difference of (ii) and (i) one gets also that

(A3) $r(B_\varepsilon - B_0)$ is weakly monotonically increasing in ε.

So, roughly speaking the dissimilarity of B_ε from B_0 increases monotonically in ε.

In a second setting we count common edges: building blocks are punished by *additive* uniform penalties $\varepsilon > 0$.

Let B_0 again be a minimum solution for the original sum-type problem with weights $r(e)$. Define s on E by

$$s(e) \quad = \quad \begin{cases} 1\,, & \text{if } e \in B_0\,, \\ 0\,, & \text{if } e \notin B_0\,. \end{cases}$$

For $\varepsilon > 0$, we thus get $f_\varepsilon(B_0) = r(B_0) + \varepsilon * |B_0|$, and in general
$f_\varepsilon(B) = r(B) + \varepsilon * |B \cap B_0|$ for arbitrary B in C.
Let B_ε be an arbitrary minimum solution for f_ε. Then statement (ii) of the theorem gives that

(B1) $r(B_\varepsilon)$ is weakly monotonically increasing in ε.

From statement (i) it follows that

(B2) $|B(\varepsilon) \cap B_0|$ is weakly monotonically decreasing in ε.

So, also for the edge-counting criterion the dissimilarity of B_ε to B_0 increases monotonically in ε.

4 Exemplary Applications

The penalty method may be applied in any situation with sum-type objective. We give some examples.

4.1 Shortest Paths

E is the edge set of the underlying network or graph. Feasible sets B of edges are paths which connect a given starting point s to a destination t. A standard approach is to punish all edges which belong to a shortest s-t-path. This is the first case described in Section 3. In a 1997 version of a commercial vehicle routing program [AND 97] all segments of a shortest (or fastest) route were penalized by substituting their lengths $w(e)$ by $1.2 * w(e)$, i. e. $\varepsilon = 0.2$.

In [Ber 00] pattern recognition in satellite images was done by shortest path methods. Here $\varepsilon = 1.0$ turned out to be an appropriate choice for getting an interesting alternative pattern.

4.2 Traveling Salesman Problem

Let T_0 be a shortest tour through all n cities for given edge lengths $w(e)$. Penalizing the edges within T_0 by an additive term $+\varepsilon$ will result in alternative optima T_ε, where the number of joint edges in T_0 and T_ε decreases monotonically with ε.

Observe that the statement of monotonicity holds independently of the computational complexity of the original optimization problem (the Traveling Salesman Problem is NP-hard).

4.3 Complete Bipartite Matchings

Consider a bipartite graph with n vertices on each side and positive real-valued cost $c(i,j)$ for each possible pair (i,j). Complete matchings with small cost sum have to be found. First of all a minimum cost matching M_0 is computed. Then all pairs in this matching are punished by setting $c'(i,j) = (1+\varepsilon) * c(i,j)$ for $(i,j) \in M_0$. All other pairs keep their original cost, hence $c'(i,j) = c(i,j)$ for them. The new optimal matching M_ε is computed for the cost function c'. Observe that the monotonicity result also holds in case of negative costs $c(i,j)$.

4.4 Alignments

Two (or several) strings over some alphabet may be aligned according to a function which gives graded rewards for each possible pair (or tuple, respectively) of letters. The total value of an alignment is typically the sum of the rewards. This value is to be maximized. Alignments of RNA- and DNA-strings play a crucial role in Computational Biology. In [WE 87] a method was introduced to support the finding of alternative alignments: disliked pairings in the original optimum were punished drastically by setting down their reward to 0. The max-version of our parametric penalty method allows more fine-tuned punishing.

5 Theses and Open Problems

5.1 Theses

- ⋆ Of course the penalty method may be applied also to maximization problems instead of minimum problems (see Example 4.4 above). Here penalties would be given for instance by introducing factors $(1-\varepsilon)$ for the relevant building blocks. Monotonicity results analogous to those in Sections 2 and 3 hold.

- ⋆ As observed already in Example 4.2 above: the computational complexity of the original optimization problem does not matter.

- ⋆ The most straightforward choice is to punish exactly those building blocks e which belong to the optimal solution computed for the original problem. But it may make sense also to penalize edges which are (in some sense)

neighbouring to the optimum solution. In shortest path problems that may be for instance those edges which switch from the shortest path or those which run directly in parallel to it.

- ⋆ Penalty methods for (integer) linear programming problems are discussed in [Sch 02].
- ⋆ Good penalty parameters depend on the application. There seems to be no uniform best choice for all cases.
- ⋆ In practice it may make sense to try different parameters ε for the same problem, either statically or dynamically, for instance in a divide and conquer manner: Assume that for ε_1 the alternative is too similar to the optimum, and for $\varepsilon_2 > \varepsilon_1$ the alternative is not competitive. Then $\varepsilon = 0.5 * (\varepsilon_1 + \varepsilon_2)$ may be tried next.
- ⋆ It may also make sense to work interactively with non-uniform penalties, concentrating on those parts of the optimum solution which are disliked. Such approaches may be practically very successful without leading to smooth theoretical results.

5.2 Open Problems

- ⋆ The situation where more than one alternative has to be generated for a fixed penalty parameter is not well understood, yet. See also the papers [AW 99a] and [AW 99b] which demonstrate analogous difficulties in case of k-best optimization under distance constraints.
- ⋆ Good automatic procedures for finding appropriate penalty parameters are still missing.
- ⋆ Good automatic procedures for finding appropriate neighbourhood sets of the optimum B_0 to be punished are still missing.

6 Acknowledgements

Thanks are due to an unknown AND-programmer for his or her nice realization of the "alternative feature" in the AND'97 vehicle routing program. From Rolf Backofen we learnt about the reference [WE 87].

References

[Alt 00a] I. Althöfer. Decision support systems with multiple choice structure. In " Numbers, Information, and Complexity" (Eds. I. Althöfer, N. Cai et al). Kluwer, Dordrecht, 2000, pp. 525-540.

[Alt 00b] I. Althöfer. K-alternative algorithms for sum-type problems (extended abstract). 13. Workshop on Discrete Optimization (Holzhau), May 2000.

[AW 99a] I. Althöfer and W. Wenzel. Two-best solutions under distance constraints: the model and exemplary results for matroids. Advances in Applied Mathematics 22 (1999), 155-185.

[AW 99b] I. Althöfer and W. Wenzel. K-best solutions under distance constraints in valuated delta-matroids. Advances in Applied Mathematics 22 (1999), 381-412.

[AND 97] AND Software GmbH. Car routing program "Route Germany". Wiesbaden, 1997.

[Ber 00] F. Berger. K alternative paths instead of k shortest paths. Diploma thesis, University of Jena, Faculty of Mathematics and Computer Science, April 2000.

[Epp 90++] D. Eppstein. Internet bibliography on k-shortest path algorithms. 436 kilobytes large at May 1, 2002.
http://www.ics.uci.edu/~eppstein/bibs/kpath.bib

[NSZZ 94] L. Nguyen, R. Schwartz, Y. Zhao, and G. Zavaliagkos. Is N-best dead? In Proc. "4th ARPA Human Language Technology Workshop" (1994), 411-414.

[Sch 02] S. Schwarz. Doctoral dissertation, University of Jena, Faculty of Mathematics and Computer Science, in preparation, Autumn 2002.

[WE 87] M. S. Waterman and M. Eggert. A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons. J. Molecular Biology 197 (1987), 723-728.