A Distance-Based Information Preservation Tree Crossover for the Maximum Parsimony Problem

Adrien Goëffon, Jean-Michel Richer, and Jin-Kao Hao

LERIA - University of Angers, 2 bd Lavoisier, 49 045 Angers Cedex 01, France
{goeffon, richer, hao}@info.univ-angers.fr

Abstract. The Maximum Parsimony problem aims at reconstructing a phylogenetic tree from DNA sequences while minimizing the number of evolutionary changes. Known to be NP-complete, the MP problem has many applications. This paper introduces a Distance-based Information Preservation (DiBIP) Tree Crossover. Contrary to previous crossover operators, DiBIP uses a distance measure to characterize the semantic information of a phylogenetic tree and ensures the preservation of distance related properties between parents and offspring. The performance of DiBIP is assessed with a mimetic algorithm on a set of 28 benchmark instances from the literature. Comparisons with 3 state-of-the-art algorithms show very competitive results of the proposed approach with improvement of some previously best results found.

1 Introduction

Phylogeny can be defined as the reconstruction of the evolutionary history of a set of species identified by sequences of molecular or morphological characters [5]. The evolutionary relationships between species are represented by a tree whose leaves are labeled by the given species. Hillis *et al.* in [11] identify many applications of phylogeny like genetic evolution, taxonomy and classification, or virus detection. The general problem of inferring the most probable phylogenetic tree according to a given criterion is computationally hard.

In the past much work has been devoted to the problem of phylogeny reconstruction following 3 main approaches. *Distance methods* rely on a matrix of distances observed between species and have polynomial-time algorithms. But they are known to lack sometimes robustness.

Probabilistic methods are based on an evolution model of characters. The Maximum Likelihood (ML) provides a general framework that consists in inferring the most probable phylogeny that maximizes the likelihood of observed species. Although ML is popular for phylogenetic inference because it is considered as a robust method, it is more computationally expensive than other methods.

Cladistic methods are based on a matrix of given characters. The most well-known method of this class relies on the Maximum Parsimony (MP) criterion. Such a method aims at building a binary tree that minimizes a cost function which corresponds to the number of evolutionary changes. The cost of a tree can be computed in polynomial time. However the problem of searching for an optimal tree is NP-complete [7]. Given

T.P. Runarsson et al. (Eds.): PPSN IX, LNCS 4193, pp. 761–770, 2006.

this fact, various heuristic algorithms have been proposed, including local search [10,8], evolutionary algorithms [14,3,15], GRASP [16] and supertrees [1].

In this paper, we are interested in solving effectively the MP problem. For this purpose, we propose a new crossover scheme, called Distance-Based Information Preservation (DiBIP) Tree Crossover. DiBIP is fundamentally different from existing crossover operators. It ensures the transmission of useful information from parents to offspring. Benchmarking results are reported and compared with some best known and performing algorithms for the MP problem.

2 The Maximum Parsimony Problem

Given a multiple alignment of a set S of n sequences of length k characters, the aim of the Maximum Parsimony problem is to find a phylogenetic tree that minimizes the number of changes (mutations) between sequences. Each leaf of the tree is associated to one of the n species and the cost (number of mutations) of the overall tree can be estimated by building sequences of parsimony from the leaves to the root of the tree. More precisely we have the following definitions:

Definition 1 (Sequence of parsimony). *Given 2 sequences S_1 and S_2 of length k such that $S_1 =< x_1, \cdots, x_k >$, $S_2 =< y_1, \cdots, y_k >$ with $\forall i \in \{1..k\}, x_i, y_i$ belong to the power set $\mathcal{P}(\Sigma)$, where Σ is the set of possible characters, the sequence of parsimony of S_1 and S_2, noted $F(S_1, S_2) =< z_1, \cdots, z_k >$ is obtained by (see Fitch [6]):*

$$\forall i, 1 \leq i \leq k, z_i = \begin{cases} x_i \cup y_i, \text{if } x_i \cap y_i = \emptyset \\ x_i \cap y_i, \text{otherwise} \end{cases}$$

The cost of the sequence of parsimony is defined by:

$$\phi(F(S_1, S_2)) = \sum_{i=1}^{k} c_i \quad \text{where} \quad c_i = \begin{cases} 1, \text{if } x_i \cap y_i = \emptyset \\ 0, \text{otherwise} \end{cases}$$

Definition 2 (Binary Tree of Parsimony). *Let S be a set of n aligned sequences of length k where each character of the sequence is expressed over a given alphabet Σ. Let $T = (V, E)$ be a binary tree, where $V = \{v_1, \ldots, v_r\}$ is the set of nodes and $E \subseteq \{(u, v)/u, v \in V\}$ is the set of edges. T is called a binary tree of parsimony of S if there exist $r = 2 \times n - 1$ nodes partitioned in 2 subsets:*

- *a set of internal nodes I composed of $n - 1$ nodes each having 2 descendants and being labeled by a (hypothetical) sequence of parsimony of the 2 descendants,*
- *a set of leaves L composed of n nodes with no descendant, bijectively labeled by the sequences of S.*

Definition 3 (Cost of a Tree of Parsimony). *Let T be a binary tree of parsimony of a set of sequences S. The cost (or score) of T, $\phi(T)$ is equal to $\sum \phi(S_w), \forall w \in I$.*

Definition 4 (Maximum Parsimony Problem). *Given a set S of n sequences of length k, expressed over an alphabet Σ, find the most parsimonious tree T of S such that the score of parsimony of T is minimum.*

For a set of sequences S, there are $\prod_{i=3}^{|S|}(2i - 3)$ possible parsimony trees. The MP problem is thus a highly combinatorial search problem.

3 Crossover Operators for MP and Trees

The literature describes several evolutionary algorithms for phylogenetic reconstruction: for instance [13,12] for the ML problem, [14,2,3,15] for the MP problem and [4] for distance-based phylogenetic approaches. Notice that conventional subtree crossover operators used in tree-based genetic programming are not directly applicable here.

Tree crossover operators designed for inferring phylogenetic trees often follow the subtree cutting-and-regrafting strategy. Generally, given 2 parents trees, a subtree is first selected from one parent (donor parent). Then the leaves of this subtree are deleted from the other parent (receiver parent), leading to an intermediate tree. The final child tree is obtained by reconnecting the subtree from the donor parent to a merge point of the intermediate tree. Obviously, exchanging the donor and receiver parents can lead to a second child. Fig. 1 shows an example with fourteen species, where the subtree $(B, (L, N))$ is taken from parent 1 and reinserted in parent 2 between the root and the subtree $((F, J), M)$ after deleting the 3 leaves $(B, L$ and $N)$ from parent 2.

With such a crossover strategy, only partial information is transmitted from parents to offspring. For instance, in the above example, a subtree with 3 leaves (out of fourteen) of the donor tree is passed on to the child. In one sense, only a small portion of information of the donor is transmitted while a large portion of information related to the eleven other species of the donor tree is lost during the crossover operation. In

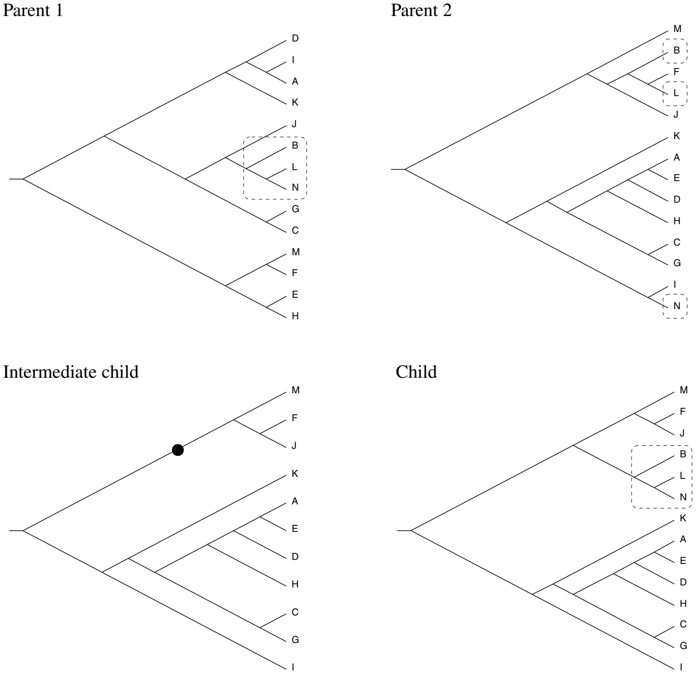


Fig. 1. Example of commonly used crossover

the next section, we introduce a Distance-Based Information Preservation crossover scheme, which ensures a global combination and transmission of information during crossover operations.

4 Distance-Based Information Preservation (DiBIP) Crossover

4.1 General Scheme

Our Distance-Based Information Preservation (DiBIP) crossover scheme aims to preserve representative properties of parents in terms of *distance* between species. The general approach can be summarized as a three-steps procedure: 1) calculate a distance matrix for each parent tree, 2) combine the matrices of the 2 parents to get a third matrix, 3) create a child tree from the previous third matrix.

In order to give a formal description of the DiBIP crossover scheme, we first introduce some notations.

- T_1 and T_2 represent 2 trees used for the crossover operation;
- δ_T is a distance metric to measure the distance of each pair of species of a tree T;
- $\varDelta : \mathcal{T} \rightarrow \mathcal{D}$ is a tree-to-distance operator to obtain a distance matrix from a tree;
- $\oplus : \mathcal{D} \times \mathcal{D} \rightarrow \mathcal{D}$ is a matrix operator to combine 2 distance matrices to produce a new distance matrix;
- $\varLambda : \mathcal{D} \rightarrow \mathcal{T}$ is a distance-to-tree operator to construct a tree from a given distance matrix.

Given these notations, the general DiBIP crossover scheme can be described with the procedure shown in Algorithm 1.

Algorithm 1 The general DiBIP crossover scheme

```
Input: $T_1, T_2, \delta_T, \varDelta, \oplus, \varLambda$
Output: A child tree $T^*$
  1. Apply the tree-to-distance operator $\varDelta$ to each parent tree $T_i$ ($i$=1,2) to obtain the corresponding distance matrix $D_i = \varDelta(T_i)$ ($i$=1,2);
  2. Apply the matrix operator $\oplus$ to $D_1$ and $D_2$ to obtain $D^*$: $D^* = D_1 \oplus D_2$;
  3. Apply the distance-to-tree operator $\varLambda$ to $D^*$ to obtain a child tree: $T^* = \varLambda(D^*)$.
```

This general scheme gives rise to several comments. First, the distance measure should be ideally correlated to the evolutionary changes between species. For instance, 2 species separated in the tree by a small number of evolutionary changes should have a smaller distance than 2 species separated by a large number of changes. A minimal requirement for the distance measure would make the measure topology dependent. In this sense, the length of the elementary path between 2 species is such a possible example. On the other hand, the conventional Hamming distance is not applicable here because this metric is totally independent of tree topologies.

Second, since we want to preserve representative properties of the parents during the crossover operation, a valid matrix operator $\oplus$ should meet some specific requirements meaningful to the MP problem and help to transmit good properties shared by both parents to the child. For instance, if a pair of species (a,b) is closer than another pair (c,d) in both parents, then this distance property should be conserved by the crossover process and transmitted to the resulting child. More generally, let (a,b) and (c,d) be 2 pairs of species, D_1 and D_2 the distance matrix of 2 trees T_1 and T_2, and $\lhd \in \{<, =, >\}$ a relation, then the following condition, called relation preservation property, should be verified:

$$(D_1(a,b) \lhd D_1(c,d)) \wedge (D_2(a,b) \lhd D_2(c,d)) \Rightarrow (D^*(a,b) \lhd D^*(c,d))$$

For example, let us consider the operation $\oplus$ such that for a pair of species (i,j), $(D_1 \oplus D_2)(i,j) = \alpha.\min\{D_1(i,j), D_2(i,j)\} + (1-\alpha).\max\{D_1(i,j), D_2(i,j)\}$ with $\alpha \in [0,1]$. It is easy to verify that this defines indeed a valid $\oplus$ operator. Moreover, this definition offers in fact many possibilities and seems particularly relevant to the MP problem. For instance, the arithmetic average ($\alpha = 0.5$) and the max operator $\max$ ($\alpha = 0$) are 2 special cases. At last, let us mention that the arithmetic addition $+$ is another simple valid $\oplus$ operator.

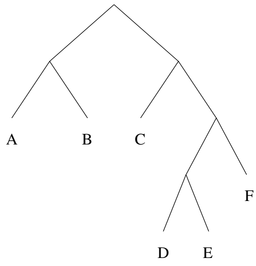
Finally, one may notice that Λ is not Δ^{-1} and the distance matrix $\Delta(T^*)$ is in general different from D^*.

To summarize, the proposed DiBIP crossover scheme is fundamentally different from conventional tree crossover operators. From this scheme, one can derive a concrete DiBIP crossover operator by defining a distance metric δ_T and instantiating the following 3 operators: Δ, $\oplus$ and Λ.

4.2 Application of the DiBIP Crossover to the MP Problem

In order to show how the above DiBIP crossover scheme is applied to the MP problem, we devise a concrete DiBIP operator by making the following choices. The distance measure δ_T between 2 species i and j is defined by the length of the elementary path between the respective ascendants of i and j, minus 1 if the path contains the root of the tree T. Since the position of the root has no effect on the parsimony score, this element must not affect the distance matrix. Fig. 2 shows a tree and the resulting distance matrix according to δ_T.

As to the matrix operator $\oplus$, we simply used the addition $+$ such that $D(i,j) = D_1(i,j) + D_2(i,j)$. Notice that this operator satisfies the relation preservation property mentioned in the previous section. Finally, the distance-to-tree operator Λ is a



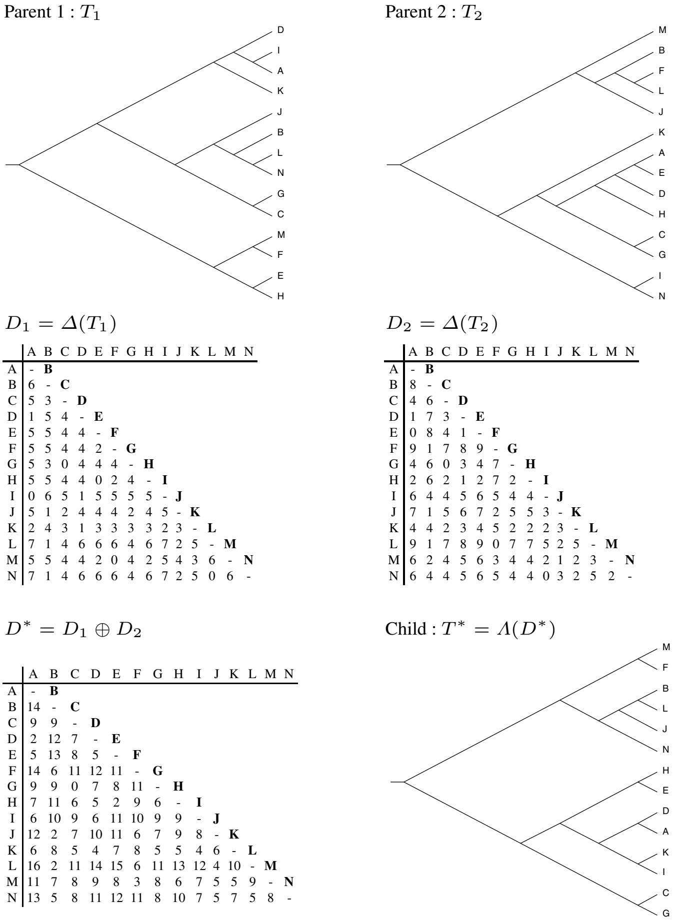
$$\begin{array}{c|ccccc|}
 & A & B & C & D & E \\
\hline
B & 0 & & & & \\
C & 1 & 1 & & & \\
D & 3 & 3 & 2 & & \\
E & 3 & 3 & 2 & 0 & \\
F & 2 & 2 & 1 & 1 & 1
\end{array}$$

Fig. 2. Distances measured using the length of elementary path between 2 species

non-deterministic variant of the well-known UPGMA method [17]. Fig. 3 shows an application of this crossover operator. One notices that the closeness of species in both parents is conserved in the child. This observation applies equally to distant species.

4.3 Complexity of the DiBIP Crossover Operator

For a given tree T, $\Delta(T)$ can be done in $\Theta(n^2 \log_2(n))$ time with n being the number of the leaves (species). This calculation is performed only once for each inferred tree even if the same tree can be used several times by the crossover operation. This is simply done by recording the corresponding distance matrix. The matrix addition using $+$ as well as the distance-to-tree operation with UPGMA have time complexity of $\Theta(n^2)$. Consequently, the crossover operator has a total time complexity of $\Theta(n^2 \log_2(n))$.



Parent 1 : T_1

Parent 2 : T_2

$D_1 = \Delta(T_1)$

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A	-	**B**												
B	6	-	**C**											
C	5	3	-	**D**										
D	1	5	4	-	**E**									
E	5	5	4	4	-	**F**								
F	5	5	4	4	2	-	**G**							
G	5	3	0	4	4	4	-	**H**						
H	5	5	4	4	0	2	4	-	**I**					
I	0	6	5	1	5	5	5	5	-	**J**				
J	5	1	2	4	4	4	2	4	5	-	**K**			
K	2	4	3	1	3	3	3	3	2	3	-	**L**		
L	7	1	4	6	6	6	4	6	7	2	5	-	**M**	
M	5	5	4	4	2	0	4	2	5	4	3	6	-	**N**
N	7	1	4	6	6	6	4	6	7	2	5	0	6	-

$D_2 = \Delta(T_2)$

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A	-	**B**												
B	8	-	**C**											
C	4	6	-	**D**										
D	1	7	3	-	**E**									
E	0	8	4	1	-	**F**								
F	9	1	7	8	9	-	**G**							
G	4	6	0	3	4	7	-	**H**						
H	2	6	2	1	2	7	2	-	**I**					
I	6	4	4	5	6	5	4	4	-	**J**				
J	7	1	5	6	7	2	5	5	3	-	**K**			
K	4	4	2	3	4	5	2	2	2	3	-	**L**		
L	9	1	7	8	9	0	7	7	5	2	5	-	**M**	
M	6	2	4	5	6	3	4	4	2	1	2	3	-	**N**
N	6	4	4	5	6	5	4	4	0	3	2	5	2	-

$D^* = D_1 \oplus D_2$

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A	-	**B**												
B	14	-	**C**											
C	9	9	-	**D**										
D	2	12	7	-	**E**									
E	5	13	8	5	-	**F**								
F	14	6	11	12	11	-	**G**							
G	9	9	0	7	8	11	-	**H**						
H	7	11	6	5	2	9	6	-	**I**					
I	6	10	9	6	11	10	9	9	-	**J**				
J	12	2	7	10	11	6	7	9	8	-	**K**			
K	6	8	5	4	7	8	5	5	4	6	-	**L**		
L	16	2	11	14	15	6	11	13	12	4	10	-	**M**	
M	11	7	8	9	8	3	8	6	7	5	5	9	-	**N**
N	13	5	8	11	12	11	8	10	7	5	7	5	8	-

Child : $T^* = \Lambda(D^*)$

Fig. 3. Application of the DiBIP Tree Crossover

5 A Hybrid Genetic Local Search Algorithm for the MP Problem

5.1 General Procedure

HYDRA (for HYbrid Distance Recombination Algorithm) is a mimetic algorithm that combines a genetic algorithm using the DiBIP crossover operator and a local search algorithm called DPN (*Descent with Progressive Neighborhood*) [9]. The HYDRA algorithm (see Algorithm 2) starts by randomly generating an initial population where each individual is an inferred phylogenetic tree (*GeneratePopulation*). Then, the algorithm enters an iterative process. At each step 2 individuals (parents) are chosen in the population (*ChooseParents*) and recombined (*DiBIP* crossover) to obtain a new individual (child). The DPN local search algorithm is applied to improve the child during l iterations. The improved child is finally added to the population under insertion conditions. This process is repeated until the stop condition is met, usually when a maximum number of iterations Max_{iter} has been reached or when the computation time exceeds a maximum duration Max_{time}.

Algorithm 2 Hybrid genetic local search algorithm (HYDRA) for the MP problem

```
Input: $A$ : an alignment of sequences, $N$ : the size of the GA population, $l$ : the number of local search iterations
Output: The most parsimonious tree found
P = GeneratePopulation($A,N$)
While not StopCondition() do
    $(T_1, T_2)$ = ChooseParents($P$)
    $T$ = DiBIP($T_1, T_2$)
    $T$ = DPN($T$,$l$)
    $P$ = Replace($P$,$T$)
return the best tree found
```

The function *ChooseParents* operates with a tournament selection strategy. Two groups of 20% of the individuals are constituted. Two individuals that represent the best individuals of each group are selected for the crossover operation. The *Replace* function inserts the child tree T into the population P and removes from P the older individual (insertion condition). Notice that other selection strategies and insertion conditions may be defined.

5.2 Local Search: Descent with Progressive Neighborhood

The DPN procedure used in HYDRA for local improvement is a basic descent algorithm using a Progressive Neighborhood [9]. DPN considers a large size neighborhood at the beginning of the search which progressively shrinks when the search goes on. The basic neighborhood used in DPN is the well-known SPR (*Subtree Pruning and Regrafting*) neighborhood which cuts a subtree and reinserts it elsewhere. To control gradually the size of the neighborhood, DPN introduces a parameter $d_{\max}$ which fixes the maximum allowed distance between the root of the detached subtree and the position where it is reconnected. In practice, $d_{\max}$ is set to a maximum value at the beginning of the search in order to allow any SPR move. Then this value is progressively reduced until it becomes equal to 1 which corresponds to the much smaller neighborhood NNI (*Nearest Neighbor Interchange*).

6 Experimentations

6.1 Competing Algorithms and Benchmarks

In this section, we compare the mimetic HYDRA algorithm based on the DiBIP crossover operator with 3 highly effective MP algorithms: an evolutionary algorithm [15], the GRASP+VND [16] method and the software TNT [10]. GRASP+VND is a combined application of 2 well-known metaheuristics GRASP and VND to the MP problem. TNT (Tree analysis using New Technology) is probably the fastest and one of the most effective parsimony analysis program. TNT uses many search strategies such as tree recombinations, local search and supertrees.

The benchmark instances used here come from [15] and [16] and represent 28 instances: 8 obtained from real data and 20 randomly generated instances (TST 01 to 20). For these instances, the best results found in the literature are reported in [15,16].

6.2 Computational Results

HYDRA uses a population of size $N = 30$, a maximum of 50,000 iterations for each DPN run and ends after 300 seconds. The algorithm is coded in C++ and compiled with *gcc* using the optimization flag `-O3`. It is run sequentially onto a cluster of 10 nodes, each having a Xeon 2 GHz BiProcessor with 1 Gb of RAM. Like in [15], the HYDRA algorithm is run 10 times for each instance.

Results are printed on Tables 1 and 2, where ϕ_b, ϕ_m and *CO* respectively represent the best score obtained, the average score and the average number of crossover operations over 10 runs. For random instances, *diff* is the improvement of the score obtained in comparison to the best known scores. The reported results of [15,16] are taken from these 2 papers while the results of TNT are obtained by us using the default parameters.

Table 1. Results on real instances

Instance			Hydra			[15]		[16]	TNT
Name	n	k	ϕ_b	ϕ_m	*CO*	ϕ_b	ϕ_m	ϕ_b	ϕ_b
GRIS	47	93	**172**	172.0	*4012*	**172**	172.0	**172**	**172**
ANGI	49	59	**216**	216.0	*1658*	**216**	216.0	**216**	217
TENU	56	179	**682**	682.0	*812*	**682**	682.0	**682**	**682**
ETHE	58	86	**372**	372.0	*2392*	**372**	372.4	**372**	373
ROPA	75	82	**325**	326.2	*1519*	**325**	325.8	**325**	327
GOLO	77	97	**496**	496.0	*2068*	**496**	496.2	**496**	501
SCHU	113	146	**759**	759.0	*669*	**759**	759.2	**759**	761
CARP	117	110	**548**	548.9	*815*	**548**	548.6	**548**	550
Average computation time (s)			300			1000		33789	< 1

Table 2. Results on random instances

Instance			Hydra			[15]		[16]	TNT	
Name	n	k	ϕ_b	ϕ_m	*CO*	ϕ_b	ϕ_m	ϕ_b	ϕ_b	*diff*
TST01	45	61	**545**	547.1	*2548*	549	549.6	551	554	-4
TST02	47	151	**1354**	1358.7	*775*	1358	1363.6	1364	1380	-4
TST03	49	111	**834**	837.3	*1012*	838	840.6	845	849	-4
TST04	50	97	**590**	591.2	*1064*	592	595.0	598	603	-2
TST05	52	75	**789**	792.6	*1458*	790	794.0	797	805	-1
TST06	54	65	**597**	599.3	*1491*	603	605.4	609	612	-6
TST07	56	143	**1271**	1275.5	*548*	1276	1280,6	1291	1300	-5
TST08	57	119	**853**	857.2	*666*	863	867.4	870	889	-11
TST09	59	93	**1146**	1149.5	*906*	1150	1154.2	1152	1167	-4
TST10	60	71	**721**	723.7	*1168*	725	728.6	733.0	740	-4
TST11	62	63	**544**	546.2	*1237*	**544**	546,8	553	564	0
TST12	64	147	**1218**	1224.1	*408*	1229	1233.0	1243	1250	-11
TST13	65	113	**1523**	1526.4	*660*	1526	1530.6	1532	1538	-3
TST14	67	99	**1167**	1171.7	*683*	1174	1177,4	1177	1194	-7
TST15	69	77	**757**	760.1	*792*	765	766.4	774	783	-8
TST16	70	69	**532**	535.5	*865*	545	547.6	551	552	-13
TST17	71	159	**2460**	2467.1	*360*	2468	2470.8	2468	2485	-8
TST18	73	117	**1529**	1533.8	*473*	1542	1548.2	1554	1571	-13
TST19	74	95	**1019**	1021.6	*601*	1028	1033.0	1036	1037	-9
TST20	75	79	**665**	668.5	*720*	676	678.8	682	693	-11
Average computation time (s)			300			1000		1982	< 1	

From Tables 1 and 2, one observes that for the 8 real instances HYDRA consistently obtains the previously best results reported in [15,16] but with much shorter computation time. Also observe the robustness of HYDRA which, for 6 of 8 instances, has found the best score for each run. The effectiveness of HYDRA is better observed on the set of random instances. Indeed, for 19 instances out of 20, the previously published best scores are improved.

Let us mention that when the number of runs is increased to 100, HYDRA obtains still better results for 11 of the 20 random instances (up to 5 units). HYDRA is also compared with its local search component DPN alone, showing clearly better results on the tested instances (not shown here).

7 Conclusion

In this paper, we have introduced the Distance-Based Information Preservation (DiBIP) crossover, a new crossover scheme for inferring phylogenetic trees. The key idea is to use a distance matrix to characterize each inferred tree. Consequently, 2 trees can be easily combined by an operation on 2 distance matrices. Contrary to existing crossover mechanisms, the DiBIP crossover scheme offers a simple and natural way to ensure a global information combination and transmission during the cross-overing operation.

The practical usefulness of the DiBIP crossover scheme for the Maximum Parsimony problem is assessed within a mimetic algorithm. Comparisons with 3 state-of-the-art algorithms on a set of 28 (real and randomly generated) benchmark instances show very competitive results of our approach. Indeed, for the real instances, the best known score are systematically found rapidly and consistently. The most remarkable results concern the random instances for which we can improve 19 (out of 20) best scores known today within 5 minutes of CPU time.

Acknowledgments. This work is partially supported by the French Ouest Genopole®. We thank the reviewers of the paper for their careful reading and useful comments and Dr. Celso C. Ribeiro for providing us with the dataset used in the paper.

References

1. O. R. P. Bininda-Emonds (Ed.). Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life. Springer, 2004.
2. C. B. Congdon. Gaphyl: An Evolutionary Algorithms Approach For The Study Of Natural Evolution. *Proceedings of the 6th Joint Conference on Information Science*, 2002.
3. C. B. Congdon and K. J. Septor. Phylogenetic trees using evolutionary search: Initial progress in extending gaphyl to work with genetic data. *Proc. of the 2003 Congress on Evolutionary Computation*, pp.320-326, IEEE press, 2003.
4. C. Cotta and P. Moscato. Inferring Phylogenetic Trees Using Evolutionary Algorithms. *Lecture Notes in Computer Science*, 2439:720-729, Springer, 2002.
5. J. Felsenstein. Inferring phylogenies. *Sinauer Associates*, 2003.
6. W. Fitch. Towards defining course of evolution: minimum change for a specified tree topology. *Systematic Zoology* 20:406-416, 1971.
7. L. R. Foulds and R. L. Graham. The Steiner problem in phylogeny is NP-complete. *Advances in Applied Mathematics* 3:43-49, 1982.
8. A. Goëffon, J.-M. Richer and J.-K. Hao. Local search for the maximum parsimony problem. *Lecture Notes in Computer Science* 3612:678-683, Springer, 2005.
9. A. Goëffon, J.-M. Richer and J.-K. Hao. Progressive Tree Neighborhood applied to the Maximum Parsimony Problem. Submitted to *IEEE Transactions on Computational Biology and Bioinformatics*, 2006.
10. P. A. Goloboff, J. S. Farris and K. Nixon. TNT: Tree Analysis Using New Technology, 2003.
11. D. Hillis, C. Moritz and B. Mable. Molecular Systematics, Sinauer, Boston, 1996.
12. P. O. Lewis. A genetic algorithm for maximum-likelihood phylogeny inference using nucleotide sequence data. *Molecular Biolology and Evolution*, 15(3):277-283, 1998.
13. H. Matsuda. Protein Phylogenetic Inference Using Maximum Likelihood With A Genetic Algorithm *Prof. of Pacific Symposium on Biocomputing*, pp.512-536, 1996.
14. A. Moilanen. Searching for Most Parsimonious Trees with Simulated Evolutionary Optimization. *Cladistics* 15(1):39-50, 1998.
15. C. C. Ribeiro and D. S. Vianna. A genetic algorithm for the phylogeny problem using an optimized crossover strategy based on path-relinking. *Proc. of 2nd Bresil Workshop on Bioinformatics*, pp.97-102, 2003.
16. C. C. Ribeiro and D. S. Vianna. A GRASP/VND heuristic for the phylogeny problem using a new neighborhood structure. *International Transactions in Operational Research* 12:1-14, 2005.
17. Sneath, P. H. A., and R. R. Sokal. 1973. Numerical taxonomy. W. H. Freeman and Co., San Francisco, California