*INFORMS Journal on Computing*
Vol. 16, No. 3, Summer 2004, pp. 232–240
ISSN 0899-1499 | EISSN 1526-5528 | 04 | 1603 | 0232

DOI 10.1287/ijoc.1030.0049

# DNA Sequencing—Tabu and Scatter Search Combined

Jacek Błażewicz
Institute of Computing Science, Poznań University of Technology, Piotrowo 3A, 60-965 Poznań, Poland, and
Institute of Bioorganic Chemistry, Polish Academy of Sciences, Noskowskiego 12, 61-704 Poznań, Poland, blazewic@put.poznan.pl

Fred Glover
Leeds School of Business, University of Colorado, Boulder, Colorado 80309-0419,
USA, fred.glover@colorado.edu

Marta Kasprzak
Institute of Computing Science, Poznań University of Technology, Piotrowo 3A, 60-965 Poznań, Poland, and
Institute of Bioorganic Chemistry, Polish Academy of Sciences, Noskowskiego 12, 61-704 Poznań, Poland, marta@cs.put.poznan.pl

In this paper, a tabu-search algorithm enhanced by scatter search is presented. The algorithm solves the DNA sequencing problem with negative and positive errors, yielding outcomes of high quality. We compare the new method with two other metaheuristic approaches: a previous tabu-search method and a hybrid genetic algorithm, and also with an old branch-and-bound approach.

*Key words*: heuristics; integer programming; computational molecular biology
*History*: Accepted by Michel Gendreau; received April 2001; revised April 2002, December 2002, May 2003; accepted July 2003.

## 1. Biochemical Preliminaries and Problem Formulation

*DNA sequencing* is one of the most important problems in computational molecular biology. The goal is to determine a sequence of nucleotides of a DNA fragment. Such a fragment is usually written as a sequence of the letters A, C, G, and T, representing four nucleotides composing the fragment, i.e., adenine, cytosine, guanine, and thymine, respectively. A short sequence of nucleotides is called an *oligonucleotide*. The sequencing process uses as input data a set of oligonucleotides of equal length, which are subsequences of one strand of the examined DNA fragment, and are derived from a hybridization experiment. Next, an original sequence of a known length is reconstructed, taking advantage of the fact that the oligonucleotides overlap one another.

In the *hybridization experiment* (Bains and Smith 1988, Lysov et al. 1988, Southern 1988, Drmanac et al. 1989), a complete oligonucleotide library is compared with many copies of one strand of the examined DNA fragment. The library consists of all ($4^l$) short one-strand DNA fragments of length $l$. In order to use the library, fragments are constructed in a special way on a *DNA chip* (Southern 1988, Fodor et al. 1991, Pease et al. 1994), where each element of the library has unique coordinates of the chip. During the hybridization reaction, copies of the longer DNA fragment join to oligonucleotides from the library in their complementary locations. Then, as a result of reading a fluorescent image of the chip, one obtains a set of oligonucleotides that are subfragments of the examined DNA fragment. This set is named the *spectrum*.

If the hybridization experiment were executed without errors, then the spectrum would be *ideal*, i.e., it would contain only all subsequences of length $l$ of the original sequence of the known length $n$. In this case, the spectrum consists of $n-l+1$ elements and to reconstruct the original sequence one must find an order of spectrum elements such that neighboring elements always overlap on $l-1$ nucleotides (see Example 1). There are several exact methods for solving the DNA sequencing problem with the ideal spectrum, described for example in Bains and Smith (1988), Lysov et al. (1988), or in Drmanac et al. (1989), but only the one proposed in Pevzner (1989) works in polynomial time.

Example 1. Suppose the original sequence to be found is ACTCTGG, $n = 7$. In the hybridization experiment one can use, for example, the complete library of oligonucleotides of length $l = 3$, composed of the following $4^3 = 64$ oligonucleotides: {AAA, AAC, AAG, AAT, ACA, ..., TTG, TTT}. As a result of the experiment performed without errors one obtains the ideal spectrum for this sequence, containing all

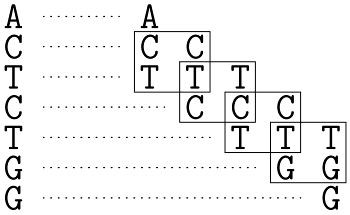

**Figure 1** The Reconstruction of the Original Sequence from the Ideal Spectrum

three-letter substrings of the original sequence: {ACT, CTC, CTG, TCT, TGG}. The reconstruction of the sequence consists of finding such an order of the spectrum elements, where each pair of neighboring elements overlaps on $l-1=2$ letters. The only possible solution for the example is presented in Figure 1. The overlapping letters of all the neighboring pairs have been framed. □

However, the hybridization experiment usually produces errors in the spectrum. There are two types of *errors*: *negative* ones, i.e., missing oligonucleotides in the spectrum, and *positive* ones, which are erroneous oligonucleotides. Every repetition of an oligonucleotide within the sequence is treated as a negative error, since the hybridization experiment cannot detect the number of occurrences of oligonucleotides in the sequence (it checks only for their presence). The presence of negative errors forces overlapping between some neighboring oligonucleotides in a sequence on fewer than $l-1$ letters. The presence of positive errors in the spectrum forces some oligonucleotides to be rejected during the reconstruction process. The existence of errors in the DNA sequencing results in strongly NP-hard combinatorial problems (Błażewicz and Kasprzak 2003). There exist exact and heuristic methods assuming errors in the spectrum, but almost all of them consider a reduced model of errors (Pevzner 1989, Drmanac et al. 1991, Lipshutz 1993, Hagstrom et al. 1994, Błażewicz et al. 1997, Halperin et al. 2002). The only exact method for the DNA-sequencing problem that allows for any type of errors and requires no additional information about the spectrum was presented in Błażewicz et al. (1999b). It generates solutions composed of a maximal number of spectrum elements (a version of the selective traveling-salesman problem), which leads to the reconstruction of the original sequences (see Example 2). The same criterion function has been used in metaheuristic methods for the problem with the most general model of errors (Błażewicz et al. 1999a, 2000, 2002). The problem can be formulated as follows.

*DNA sequencing with negative and positive errors—search version*

*Instance*: Set $S$ (spectrum) of words of equal length $l$ over the alphabet {A, C, G, T}, the length $n$ of an original sequence.

*Goal*: Find a sequence of length $\leq n$ containing the maximal number of elements of $S$.

The mathematical-programming formulation of the problem is given below.

$$\text{Maximize} \quad \sum_{i=1}^{z}\sum_{j=1}^{z} b_{ij}+1 \tag{1}$$

$$\text{subject to} \quad \sum_{i=1}^{z} b_{ik}\leq 1, \quad k=1,\ldots,z \tag{2}$$

$$\sum_{i=1}^{z} b_{ki}\leq 1, \quad k=1,\ldots,z \tag{3}$$

$$\sum_{k=1}^{z}\left(\left|\sum_{i=1}^{z} b_{ki}-\sum_{j=1}^{z} b_{jk}\right|\right)=2 \tag{4}$$

$$\sum_{s_k\in S'}\left(\sum_{s_i\in S'} b_{ik}\cdot\sum_{s_j\in S'} b_{kj}\right)<|S'|,$$

$$\forall\, S'\subset S,\; S'\neq\varnothing \tag{5}$$

$$\sum_{i=1}^{z}\sum_{j=1}^{z} c_{ij}b_{ij}\leq n-l \tag{6}$$

where:

$S=$ the spectrum,

$s_i=$ an element of the spectrum,

$z=$ the cardinality of the spectrum,

$n=$ the length of an original sequence,

$l=$ the length of a spectrum element,

$b_{ij}=$ a boolean variable equal to 1 if element $s_i$ is the immediate predecessor of element $s_j$ in a solution; otherwise it is equal to 0,

$c_{ij}=$ a cost of a connection of element $s_i$ with element $s_j$ equal to the difference between $l$ and a number of letters of the common part of $s_i$ and $s_j$ coming from their maximal overlapping.

The maximized *criterion function* (1) is equivalent to the number of spectrum elements composing the solution. Inequalities (2) and (3) guarantee that every element of the spectrum will be joined in the solution with, respectively, at most one element from the left side and at most one element from the right side. The addition of equation (4) ensures that exactly two elements connected from only one side with other elements will appear in the solution. These elements will constitute the beginning and the end of the reconstructed sequence. Supplying the above formulation with (5) allows to eliminate the solutions including subcycles of elements (when an element in the solution is simultaneously a successor and the immediate predecessor of another element from the solution). According to (6) the length of the reconstructed sequence cannot exceed its known length (the length can be shorter, for example, in case of negative errors appearing at the end of the sequence).

Example 2. To make the problem of the DNA sequencing computationally hard (on the basis of the spectrum from Example 1), we introduce some errors into the spectrum. Let the negative error be CTC, and the positive errors be CAA and TTG. Then the spectrum would have the following components: {ACT, CAA, CTG, TCT, TGG, TTG}. The use of the criterion function from Błażewicz et al. (1999b), i.e., the maximization of the number of spectrum elements composing the solution of length not greater than $n = 7$, would produce here the following two orders of oligonucleotides: (ACT, TCT, CTG, TGG) and (CAA, ACT, CTG, TGG), resulting in the two optimal solutions: ACTCTGG and CAACTGG, respectively. One of them is the original sequence. However, data coming from real hybridization experiments usually allow for a reconstruction of only one optimal solution. □

In this paper, we present a new metaheuristic algorithm for the DNA-sequencing problem with negative and positive errors. The algorithm is based on the tabu-search approach from Błażewicz et al. (1999a) enhanced by scatter search. The computational outcomes have been compared with the results of two other metaheuristic approaches: a previous tabu-search method (Błażewicz et al. 1999a) and a hybrid genetic algorithm (Błażewicz et al. 2002). The new results provide notable improvements, yielding sequences of very high similarity to the original sequences, despite the fact that computationally hard instances have been used in the tests, thus introducing a high percentage of errors.

## 2. The Algorithm

### 2.1. General Remarks

The algorithm proposed in this paper uses the same *criterion function* as the previous methods (Błażewicz et al. 1999a, b, 2002) for solving the DNA-sequencing problem with negative and positive errors. The goal is to maximize the number of elements from a spectrum, composing a solution being a sequence of nucleotides not longer than $n$ (it can be shorter in the case of negative errors at the end). The criterion function is justified by the fact that most of the information from the hybridization experiment is correct. Otherwise, it would be impossible to reconstruct an original sequence without additional information, which is hard to obtain. The algorithm also accepts the *general model of errors*, i.e., it assumes that any types of errors are possible in a spectrum. Thus, as the input to the algorithm we have only a spectrum (an arbitrary set of words of equal length $l$) and a value of $n$. The main scheme of the algorithm is based on tabu search (Glover and Laguna 1997), utilizing scatter search (Glover 1977, 1999) as a part of the diversification strategy.

In our approach the spectrum is represented by two data structures: an ordered list of oligonucleotides composing a current *solution*, and an unordered set of remaining oligonucleotides, called a *trash set*. At each stage of the computation, the number of elements from the list cannot be greater than the one that would produce a sequence of at most $n$ nucleotides (with maximal possible overlapping of the neighbors on the list). To satisfy this constraint, only the moves that do not lead to sequences of length greater than $n$ are considered. Such moves are called *feasible*. After restricting our attention to such moves, each solution generated during the computation is acceptable. Of course, oligonucleotides never appear more than once on the list representing the solution.

At the beginning, an *initial solution* is created by the *greedy heuristic* from Błażewicz et al. (1999b). The first oligonucleotide on the list is chosen at random, and successive elements are added according to the following rule. For each candidate oligonucleotide, consider the sum of numbers of overlapping letters (assuming maximal possible overlaps): (a) between the last element on the list and the considered oligonucleotide, and (b) between the considered oligonucleotide and its best possible successor. Then we choose the candidate to add next that gives a maximum value across all such pairs. Of course, in every step only oligonucleotides not yet used are taken into account. This heuristic, although simple, often generates solutions whose criterion function values are close to the optimum. On the other hand, a number of cases exists where the greedy heuristic does not do so well. In addition, most methods can get answers of similar quality, but the advantage of getting answers that remove the last gap between "good" and "extremely good" is very important, and thus the need for more powerful procedures arises.

Three basic types of *moves* are used: an *insertion* (a move transferring an oligonucleotide from the trash set to the solution), a *deletion* (a move transferring an oligonucleotide from the solution to the trash set), and a *shift* (a move within the solution). Actions on single oligonucleotides are seldom sufficient, so we have added moves using *clusters*. A cluster is a group of neighboring elements from the solution, linked together with overlaps on $l - 1$ letters in each case. Because insertion, deletion, or shift can change the composition of a cluster, the list of clusters is updated after every move. They are remembered as pairs of positions within the current solution: The first position identifies the beginning of a cluster, the second one identifies its end. Clusters exist only within the solution; they are broken into separate elements once transferred to the trash set. In sum, the set of moves is defined specifically as follows: insertion of an oligonucleotide, deletion of an oligonucleotide or a cluster, and shift of an oligonucleotide or a cluster. The following rules, limiting the application of the moves, have been established:

- A cluster may be shifted only if it does not break another cluster.
- Only an oligonucleotide outside a cluster may be shifted, provided it does not break any cluster.
- Only an oligonucleotide outside a cluster or constituting one of the cluster's ends may be deleted.

These rules avoid moves that would entail an extension of the computation time without affording an appreciable gain in solution quality.

### 2.2. Overview of the Tabu-Search and Scatter-Search Procedures

We first give a general description of the tabu-search and scatter-search components of our method, and then provide a summarizing pseudo-code description. In our tabu-search method, inserted or shifted oligonucleotides are remembered by storing them on the *tabu list* for a given number of iterations. The list is checked if an attempt to shift or to delete an oligonucleotide is made, and these moves are prevented if the oligonucleotide is on the list. It does not appear necessary to remember clusters shifted in order to avoid cycling, since clusters often change after a move. An element found on the tabu list may be deleted or shifted together with the cluster containing it. The element also may be deleted if there is no other feasible move. In such a case, an element that has been on the tabu list for the greatest number of iterations is chosen.

The global criterion function to be maximized is the number of spectrum elements composing the solution. On the other hand, a function that is able to compare all kinds of moves is a *condensation*, defined for each solution to be the ratio of the number of oligonucleotides from the spectrum in the solution to the number of nucleotides in the solution. If the moves were compared by the global criterion function, deletions or shifts would be used very rarely. Maximizing the condensation causes the initial solution to be transformed into a series of collections of well-matched oligonucleotides. (If a maximal value of the condensation is achieved by more than one move, the method selects the move resulting in the greatest number of elements in the solution. Consequently, insertion is the most preferred move with shifts, with deletion of an oligonucleotide and deletion of a cluster being next.) Obviously, using the condensation as the only criterion for choosing a move would lead after a number of iterations to creating a single cluster of length (in nucleotides) much less than $n$. This is why we decided to use both functions simultaneously (the global one and the condensation) during the search for a solution: The first one lengthens the current solution, and the second one condenses it. The formal dependence of these functions is shown in the pseudo-code description of the algorithm (§2.3). The above process of improving the current solution is the *intensification* part of the algorithm. The *diversification* strategy is described in the next subsections, and its elements are extending moves and restarts based on the scatter search.

**2.2.1. Uses of Frequency Memory.** *Extending moves* are feasible moves selected by the use of *frequency-based memory* instead of the condensation function. They are executed after a given number of condensing moves without improvement to the value of the global criterion function. The frequency-based memory is a tabu-search structure that remembers the number of times each element from the spectrum appears in solutions. Thus, for example, an element contained in all solutions generated so far has its frequency value equal to the number of iterations of the algorithm, and an element never used for constructing solutions has the frequency value equal to 0. There are two types of extending moves: the insertion of an oligonucleotide and the deletion of an oligonucleotide. The more highly preferred move is the insertion, and the oligonucleotide with the lowest frequency value is chosen. If no insertion is possible, the oligonucleotide of the highest frequency value is deleted from the solution.

After the execution of extending moves, the algorithm returns to the normal scheme with the condensation as the criterion function. Such a combination of condensing and extending the solution guarantees that the number of oligonucleotides will increase from some value in the initial solution to a near-optimal or even optimal value in the final one. The use of frequency-based memory forces the inclusion of elements that are not well-matched into the solution, which would be impossible using the condensation function only. Several forms of frequency-based memory are elements of more general tabu-search formulations (see, e.g., Glover and Laguna 1997), and we have used only one of them. Our choice in this case is to provide a diversification strategy for the algorithm that operates through a series of extending moves.

Diversification is also present in the procedure of *restarting* the algorithm, based on the *scatter-search approach*, described below.

**2.2.2. The Scatter-Search Component.** During a given number of cycles of condensing and extending moves, our scatter-search approach constructs a *reference set* by remembering a selected number of the best generated solutions. The reference set is used in our present method as a source to generate a new initial solution within the restart procedure. This use of scatter search to guide the restarting process is different from its customary role, which operates within the main body of the algorithm. (In this regard, our present approach embodies elements of the structured combination processes proposed in Glover 1994.)

A solution is a candidate to enter the reference set if it is better than one of the solutions in the set, i.e., it has a greater value of the global criterion function. The worst solution from the set is then deleted. To avoid the situation where a number of highly similar good solutions (e.g., differing by only one move) fill the set, the set can be updated only if at least ten moves have been executed after the last update. The greater the difference between solutions in the set, the greater the possibility of a good restart for the next intensification cycle. This restriction is not used when considering a solution better than all the solutions present in the set. (An alternative strategy for maintaining a diversifying influence within the reference set, which has proved highly effective in other settings, divides the set into two tiers, where solutions are evaluated for membership in the second tier based on their diversification properties; see, e.g., Glover et al. 2000.)

After that, we generate a solution using the greedy heuristic, in the same way as at the beginning of the algorithm. This solution replaces the worst one from the reference set, because it usually has a large global criterion-function value and differs from the ones present in the set (a next element of the diversification strategy). Then we generate on the basis of the reference set a new solution, again using the greedy heuristic. However, this time the heuristic does not operate on all possible connections between oligonucleotides from the spectrum but takes into account only the connections that are present within the solutions from the set. (An exception occurs when the current element has no successors. Then the method chooses as its successor some not-yet-used oligonucleotide.) Hence, the graph representing the connections becomes rather sparse, as opposed to the complete graph in the previous application of the heuristic. Now, as the first oligonucleotide in the solution we consider in turn all spectrum elements, and the solution having the greatest value of the global criterion function is chosen as the new initial solution for the next cycle of condensing and extending moves. At the end, all algorithm variables are set to initial values (except for the variables remembering the best solution found so far), and the next search process can start, independently of the previous ones. The number of restarts is a parameter of the algorithm and, at the end, the solution containing the largest number of oligonucleotides from the spectrum, found so far, is returned by the algorithm.

The scatter-search component succesfully enhances the previous tabu-search algorithm (Błażewicz et al. 1999a) by an evolutionary design, generating new initial solutions in the restart procedure on the basis of a population of very good solutions found between restarts. The greedy heuristic composing a new initial solution from the population is quite deterministic, in contradiction to the analogous procedure from the hybrid genetic algorithm (Błażewicz et al. 2002), and it results in better final solutions, as demonstrated in §3.

### 2.3. A Pseudo-Code Description

The algorithm is presented below in pseudo-code description.

```
• generate an initial solution
• while not all intensification stages
  performed do
    • while not all cycles of condensing and
      extending moves performed do
        • while not all condensing moves without
          improvement to the global criterion
          function value performed do
            • execute a feasible move of the
              greatest condensation value
            • if new solution is the best one
              generated so far according to
              global criterion function then
              store it
        • while not all extending moves
          performed do
            • execute insertion of an
              oligonucleotide with smallest
              frequency value; if not possible,
              execute deletion of an
              oligonucleotide with largest
              frequency value
            • if new solution is the best one
              generated so far according to
              global criterion function then
              store it
    • restart the search process using scatter
      search
• return the solution of maximum value of the
  global criterion function
```

## 3. Computational Testing and Results

In the computational experiments, the proposed algorithm has been compared with the two other metaheuristic approaches: a previous tabu-search method described in Błażewicz et al. (1999a), using only the greedy procedure to generate initial solutions, and a hybrid genetic algorithm from Błażewicz et al. (2002). Below we cite results from Błażewicz et al. (2002), comparing the two previous metaheuristics (Tables 1 and 2), and we add results of the new algorithm for the same instances, on the same computer. We additionally tested the old branch-and-bound algorithm for the same problem (Błażewicz et al. 1999b).

**Table 1** Results of the Previous Tabu-Search Algorithm

| Spectrum size | 100 | 200 | 300 | 400 | 500 |
|---|---|---|---|---|---|
| Average quality | 80.0 | 158.6 | 235.5 | 313.8 | 391.1 |
| Optimal quality | 80 | 160 | 240 | 320 | 400 |
| Optimum number | 40 | 24 | 11 | 6 | 2 |
| Average similarity score (points) | 108.4 | 184.1 | 196.6 | 229.5 | 235.1 |
| Average similarity score (%) | 99.7 | 94.0 | 81.8 | 78.1 | 73.1 |
| Average computation time (sec) | 14.1 | 60.8 | 177.7 | 258.3 | 471.5 |

The experiment was performed on a PC with a Pentium II 300 MHz processor, 256 MB RAM, and the Linux operating system. All spectra used in the experiment were derived from DNA sequences coding human proteins (taken from GenBank, National Institutes of Health, USA). The spectra contain 20% random negative errors and 20% random positive errors. Because cardinalities of the spectra vary from 100 to 500 oligonucleotides, they contain from 40 to 200 errors (in the latter case 100 randomly chosen oligonucleotides are missing and in addition 100 oligonucleotides in a spectrum are erroneous). The spectra have been sorted alphabetically, thus no information about an original order of oligonucleotides in the sequences has been kept. The size of oligonucleotides is in all cases equal to 10. The lengths of original sequences ($109 \leq n \leq 509$) and of oligonucleotides ($l = 10$) were chosen on the basis of real hybridization experiments (cf. Pease et al. 1994). However, all algorithms accept any values of $n$ and $l$, provided $l \leq n$.

The sequences produced by the methods were compared with original sequences using a classical pairwise alignment algorithm (Waterman 1995). The algorithm was called with the following parameters: a match (the same nucleotides at a given position in strings) brings a profit of 1 point, a mismatch (different nucleotides) brings a penalty of 1 point (i.e., $-1$) and a gap (a nucleotide against a space at the same position in the second string) also brings a penalty of 1 point. Therefore, the highest score (similarity) would equal the number of nucleotides in the sequences (in case the two sequences are identical) and the lowest score would equal the number of nucleotides in the longer sequence multiplied by $-1$ (in case the two sequences are completely different).

Parameters of the previous tabu-search algorithm were set to values resulting in computation times similar to those used by the hybrid genetic algorithm. The parameters of the new algorithm keep the same values to enable improvements caused by the changed restart procedure to be observed. The only new parameter is the cardinality of the reference set for scatter search. The parameters and their values are listed here:

- the number of condensing moves performed without improvement of the value of the global criterion function: 2;
- the number of extending moves: 4;
- the number of the cycles of condensing and extending moves: 300;
- the number of intensification stages (i.e., the number of restarts +1): 15 for spectra of cardinalities 100, 200, and 300 and 10 for spectra of cardinalities 400 and 500 (these values were used to obtain similar computation times for the new method and the older ones);
- the tabu tenure (a length of the tabu list): 10;
- the cardinality of the reference set used in the restarts: 8.

**Table 2** Results of the Hybrid Genetic Algorithm

| Spectrum size | 100 | 200 | 300 | 400 | 500 |
|---|---|---|---|---|---|
| Average quality | 80.0 | 159.4 | 237.6 | 315.9 | 393.2 |
| Optimal quality | 80 | 160 | 240 | 320 | 400 |
| Optimum number | 40 | 31 | 20 | 9 | 5 |
| Average similarity score (points) | 108.4 | 199.3 | 274.1 | 301.7 | 326.0 |
| Average similarity score (%) | 99.7 | 97.7 | 94.3 | 86.9 | 82.0 |
| Average computation time (sec) | 13.5 | 63.4 | 154.9 | 263.4 | 437.9 |

In Tables 1 and 2, computational results of the previous tabu-search algorithm and the hybrid genetic algorithm are presented, respectively. All entries with average values have been calculated for 40 instances, derived from 40 different sequences. The quality means the number of spectrum elements composing a solution. For the given instances, a value of the criterion function reached by the algorithm cannot exceed the optimal quality, which is the number of proper oligonucleotides in a spectrum. Below the qualities, the numbers of optimal solutions returned by the algorithm, out of 40, are shown (i.e., the numbers of instances solved optimally). Similarity scores, summed as described above, are shown as numbers of points (with maximal values from 109 to 509, respectively) and in percentages (with a maximum of 100% when the two sequences are equal).

As we see, both methods produce solutions of very high quality. However, the hybrid genetic algorithm is better than the previous tabu-search algorithm that excludes the scatter-search diversification component. For instances of cardinality 100, the algorithms returned only original sequences. Similarity values smaller than 100% are caused in that case by missing information about the last nucleotides in the sequences (negative errors). Even for large spectra with many errors of both types, the algorithms yield very good sequences. The solutions obtained have average qualities that range from 97.8% to 100% of the optimum values in Table 1 and from 98.3% to 100% of the optimum values in Table 2. Sometimes an instance has more than one optimal solution. In that case, an optimal solution returned by an algorithm can differ from the original one. Thus, similarities presented

**Table 3 Results of the New Tabu-Search Algorithm**

| Spectrum size | 100 | 200 | 300 | 400 | 500 |
|---|---|---|---|---|---|
| Average quality | 80.0 | 159.9 | 239.2 | 318.1 | 396.4 |
| Optimal quality | 80 | 160 | 240 | 320 | 400 |
| Optimum number | 40 | 38 | 31 | 21 | 18 |
| Average similarity score (points) | 108.4 | 207.6 | 273.7 | 323.9 | 361.4 |
| Average similarity score (%) | 99.7 | 99.7 | 94.3 | 89.6 | 85.5 |
| Average computation time (sec) | 14.6 | 61.7 | 178.3 | 265.7 | 474.5 |

**Table 4 How the Quality Improves with a Growing Number of Restarts**

| | 400 | | 500 | |
|---|---|---|---|---|
| Spectrum size | Average quality | Optimum number | Average quality | Optimum number |
| Stage #1 | 306.2 | 0 | 384.1 | 0 |
| Stage #2 | 313.5 | 3 | 390.6 | 7 |
| Stage #3 | 314.7 | 7 | 393.4 | 12 |
| Stage #4 | 315.9 | 10 | 394.2 | 13 |
| Stage #5 | 316.6 | 12 | 394.7 | 14 |
| Stage #6 | 317.0 | 13 | 394.8 | 15 |
| Stage #7 | 317.3 | 16 | 394.9 | 15 |
| Stage #8 | 317.6 | 18 | 395.7 | 17 |
| Stage #9 | 317.9 | 20 | 396.1 | 18 |
| Stage #10 | 318.1 | 21 | 396.4 | 18 |

in the tables are in fact lower bounds on the quality measure of the algorithms. For the hybrid genetic algorithm, the similarities of generated sequences to original ones are much larger than in the case of the earlier tabu-search method.

Table 3 presents results of tests done with the new tabu-search method. The outcomes are much better than those obtained by the previous metaheuristic approaches, which is somewhat remarkable in view of the high quality of the previous solutions, even for the computationally hardest instances. The difference is entirely due to incorporating the scatter-search approach within the restart procedure, which is the only part where the tabu-search methods differ. During a short computation time the new algorithm generated optimal solutions surprisingly often. All remaining solutions are very close to optimal, and their similarities to the original sequences are very high. The average quality varies from 99.1% (in the case of spectra of cardinalities 500, with 100 negative errors and 100 positive ones) to 100% of the optimum values.

From a practical standpoint, it is highly important that the proposed algorithm returns many more optimal solutions (the row "optimum number") than the previous methods. It can happen that a biochemical user, who would like to get a sequence reconstructed on the basis of his experiment, is interested only in obtaining the exact solution. Because the DNA sequencing problem with errors is highly complex (i.e., strongly NP-hard), this would normally be impossible using exact, exponential-time algorithms. Thus, a method that runs in polynomial time, and that often returns optimal solutions, is very valuable. In the experiment, almost all solutions being optimal as measured by the global criterion function appear to be optimal also for biochemists, because they are identical to the original sequences that provide the data (sometimes missing up to three nucleotides at the end because of negative errors). The only exceptions were two instances of cardinality 300, where the constructed sequences composed of 240 oligonucleotides differed substantially from the original sequences. The potential ambiguity of results (where two or more optimal solutions are possible) cannot be resolved without additional information about original sequences, which is not contained within spectra. Therefore, the choice of the criterion function in the algorithm has been proved to give an appropriate evaluation for the given information (which consists of only a spectrum and the length of a sequence).

Table 4 shows how the quality of solutions returned by the new algorithm improves with a growing number of restarts. This table presents results for the spectra of cardinalities 400 and 500, and for ten intensification stages, that are set for these instances. Values in the row "stage #*i*" refer to the quality of the best solutions generated until the *i*th restart of the algorithm. Values from the row "stage #10" are the final ones, the same as in Table 3.

The average qualities from Table 4 change with each restart executed by the algorithm. The most significant stage is the first one, where the greedy heuristic generates very good initial solutions, and they are further improved in the following intensification phase. Also, the first restart considerably improves the quality, yielding new solutions that are good starting points for the succeeding intensification procedure. Every next restart returns a solution not worse than in the previous restart, because the best solution found so far is remembered in the algorithm. Similar results can be observed for spectra of cardinalities 100, 200, and 300. Figure 2 shows how the numbers of optimal solutions (for 40 instances in each case) change with a growing number of restarts.

In general, the larger the number of restarts executed by the algorithm, the better the quality of the solutions generated. Therefore, another series of tests was done to determine how the method works when longer computation times are allowed. Only the hardest instances were chosen in order to allow the possibility of observing potential improvements. The time was set to about 40 minutes (in order to compare the results with the previous ones). It required the change of the number of intensification stages from ten to 50. The results of these tests are presented in Table 5 (the third column). For comparison, we

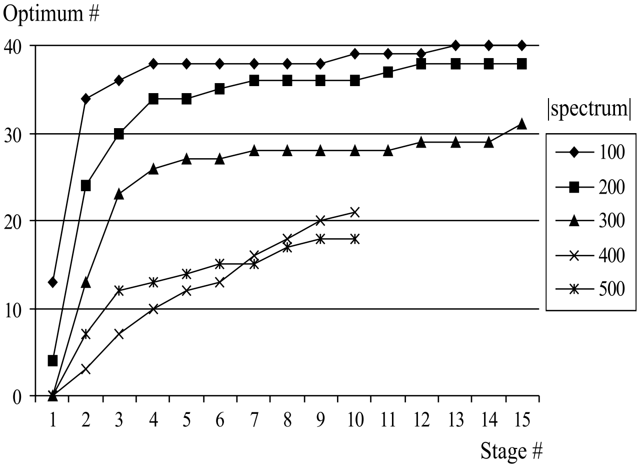

**Figure 2** Number of Optimal Solutions at Given Intensification Stage of the Algorithm

also include results cited from Błażewicz et al. (2002) for the previous tabu-search approach (the first column) and the hybrid genetic algorithm (the second column).

Again, the new algorithm performs very effectively. The number of optimal solutions generated by the algorithm is much greater than for the previous algorithms, and the average quality equals 99.6% of the optimal value. We also observe that the average quality provided by increasing the run time for the new method grew to 398.3 compared to 396.4 for the shorter length of execution, which itself is superior to the quality of 396.0 produced by the longer execution of the hybrid GA approach.

At the end, the old branch-and-bound algorithm from Błażewicz et al. (1999b) was tested on the current sets of 40 instances. The computations were done on the same computer and—as in the above tests—without knowledge of the oligonucleotide beginning the original sequences (as opposed to the tests from Błażewicz et al. 1999b). The results are presented in Table 6.

The time limit of the computations was set to one hour for every instance. As we can see, even for small instances—when spectrum size is equal to 200—the computation time of this exact, exponential-time algorithm is very high; only one instance from the set of 40 was solved within the time limit. Of course, all unbroken runs of the algorithm returned exact solutions. However, we also counted the average quality over the 40 instances, which is the mean value of the number of oligonucleotides used in building the best solutions found within the time limit. The average computation times were calculated only for instances solved within one hour (i.e., for 29 and one instances, respectively). The results from Table 6 show that the exact algorithm for the DNA-sequencing problem with both types of error returns much worse average results than do all the heuristics presented here, and in much longer computation time. The heuristics for spectrum size equal to 100 returned average quality equal to 80 in all cases (the exact algorithm returned 77.2), and for spectrum size equal to 200 they returned average qualities of 158.6, 159.4, and 159.9 (as compared with 151.3 for the branch-and-bound method). These outcomes clearly establish the high quality of the heuristic method presented here.

**Table 5** Results for Spectra of Cardinality 500, with Computation Time Set to 40 Minutes

| | Previous TS | Hybrid GA | TS + SS |
|---|---|---|---|
| Average quality | 394.1 | 396.0 | 398.3 |
| Optimal quality | 400 | 400 | 400 |
| Optimum number | 4 | 9 | 23 |
| Average similarity score (points) | 286.0 | 393.1 | 410.0 |
| Average similarity score (%) | 78.1 | 88.6 | 90.3 |

**Table 6** Results of the Old Branch-and-Bound Algorithm

| Spectrum size | 100 | 200 |
|---|---|---|
| Number of instances tested | 40 | 40 |
| Number of instances solved within 1 hour | 29 | 1 |
| Optimal quality | 80 | 160 |
| Average quality for solved instances | 80.0 | 160.0 |
| Average quality for all tested instances | 77.2 | 151.3 |
| Average computation time for solved instances (sec) | 891.2 | 1,273.0 |

## 4. Conclusion

In the paper, we have presented a tabu-search algorithm enhanced by a diversification process that embeds scatter search in the restart procedure. The algorithm solves the DNA-sequencing problem for instances that contain a large percentage of both negative and positive errors, yielding solutions of surprisingly high quality. Computational experiments were performed to compare the algorithm with two other metaheuristic approaches: a previous tabu-search method and a hybrid genetic algorithm. The new results are much better than the previous ones. During a short computation time the new algorithm often generates optimal solutions. All remaining solutions are very close to the optimum, and their similarities to original sequences are very high. Our tests also demonstrate the merit of the criterion function used in the algorithm, which measures the degree to which solutions match the original sequences.

In spite of the high quality of our results, they could be further improved. For example, one apparent opportunity would be to use an advanced multi-start method in place of the greedy heuristic as a way of generating starting solutions for the method. Moreover, the proposed method has not used any additional information about spectra or original sequences, which could be derived from biochemical experiments. For example, one could assume that the first (or last) oligonucleotide of an original sequence is known, based on the knowledge about primers used by biochemists to amplify an examined molecule in PCR reaction before sequencing. Then, sequences obtained could be made to match original ones more closely. Other information might come from databases, such as a probabilistic analysis of existing characteristic subsequences in particular genes, which could exclude several low-probable orders of oligonucleotides. However, given that such additional information is not always accessible, we have proposed a more general algorithm of wider applicability. Currently, the ratio of errors to proper oligonucleotides in the problem data is rather large. In experimental data from real applications one could expect a smaller number of errors and then the algorithm would provide even better results in these practical settings.

## Acknowledgments

This research has been partially supported by a KBN grant.

## References

Bains, W., G. C. Smith. 1988. A novel method for nucleic acid sequence determination. *J. Theoret. Biol.* **135** 303–307.

Błażewicz, J., P. Formanowicz, F. Glover, M. Kasprzak, J. Węglarz. 1999a. An improved tabu search algorithm for DNA sequencing with errors. Celso C. Ribeiro, ed. *Proc. III Metaheuristics Internat. Conf. MIC'99, Angra dos Reis, July 1999*, Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil, 69–75.

Błażewicz, J., P. Formanowicz, M. Kasprzak, W. T. Markiewicz, J. Węglarz. 1999b. DNA sequencing with positive and negative errors. *J. Comput. Biol.* **6** 113–123.

Błażewicz, J., P. Formanowicz, M. Kasprzak, W. T. Markiewicz, J. Węglarz. 2000. Tabu search for DNA sequencing with false negatives and false positives. *Eur. J. Oper. Res.* **125** 257–265.

Błażewicz, J., J. Kaczmarek, M. Kasprzak, W. T. Markiewicz, J. Węglarz. 1997. Sequential and parallel algorithms for DNA sequencing. *Comput. Appl. Biosci.* **13** 151–158.

Błażewicz, J., M. Kasprzak. 2003. Complexity of DNA sequencing by hybridization. *Theoret. Comput. Sci.* **290** 1459–1473.

Błażewicz, J., M. Kasprzak, W. Kuroczycki. 2002. Hybrid genetic algorithm for DNA sequencing with errors. *J. Heuristic* **8** 495–502.

Drmanac, R., I. Labat, I. Brukner, R. Crkvenjakov. 1989. Sequencing of megabase plus DNA by hybridization: Theory of the method. *Genomics* **4** 114–128.

Drmanac, R., I. Labat, R. Crkvenjakov. 1991. An algorithm for the DNA sequence generation from k-tuple word contents of the minimal number of random fragments. *J. Biomolecular Structure Dynam.* **8** 1085–1102.

Fodor, S. P. A., J. L. Read, M. C. Pirrung, L. Stryer, A. T. Lu, D. Solas. 1991. Light-directed, spatially addressable parallel chemical synthesis. *Science* **251** 767–773.

Glover, F. 1977. Heuristics for integer programming using surrogate constraints. *Dec. Sci.* **8** 156–166.

Glover, F. 1994. Tabu search for nonlinear and parametric optimization (with links to genetic algorithms). *Discrete Appl. Math.* **49** 231–255.

Glover, F. 1999. Scatter search and path relinking. D. Corne, M. Dorigo, F. Glover, eds. *New Ideas in Optimization*. McGraw-Hill, New York, 297–316.

Glover, F., M. Laguna. 1997. *Tabu Search*. Kluwer Academic Publishers, Boston, MA.

Glover, F., M. Laguna, R. Marti. 2000. Fundamentals of scatter search and path relinking. *Control Cybernetics* **29** 653–684.

Hagstrom, J. N., R. Hagstrom, R. Overbeek, M. Price, L. Schrage. 1994. Maximum likelihood genetic sequence reconstruction from oligo content. *Networks* **24** 297–302.

Halperin, E., S. Halperin, T. Hartman, R. Shamir. 2002. Handling long targets and errors in sequencing by hybridization. *Proc. 6th Annual Internat. Conf. Res. Comput. Molecular Biol. RECOMB*, Washington D.C., April 2002, 176–185.

Lipshutz, R. J. 1993. Likelihood DNA sequencing by hybridization. *J. Biomolecular Structure Dynam.* **11** 637–653.

Lysov, Y. P., V. L. Florentiev, A. A. Khorlin, K. R. Khrapko, V. V. Shik, A. D. Mirzabekov. 1988. Determination of the nucleotide sequence of DNA using hybridization with oligonucleotides. A new method. *Dokl. Akad. Nauk SSSR* **303** 1508–1511.

Pease, A. C., D. Solas, E. J. Sullivan, M. T. Cronin, C. P. Holmes, S. P. A. Fodor. 1994. Light-generated oligonucleotide arrays for rapid DNA sequence analysis. *Proc. National Acad. Sci. USA* **91** 5022–5026.

Pevzner, P. A. 1989. l-Tuple DNA sequencing: Computer analysis. *J. Biomolecular Structure Dynam.* **7** 63–73.

Southern, E. M. 1988. United Kingdom Patent Application GB8810400.

Waterman, M. S. 1995. *Introduction to Computational Biology. Maps, Sequences and Genomes*. Chapman & Hall, London, U.K.