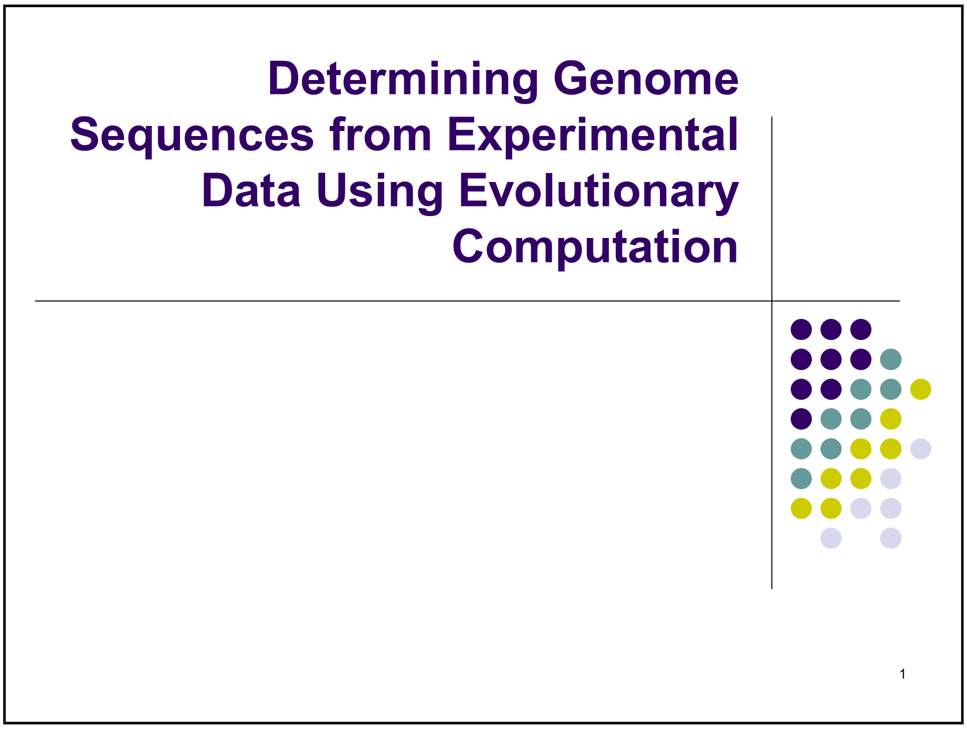
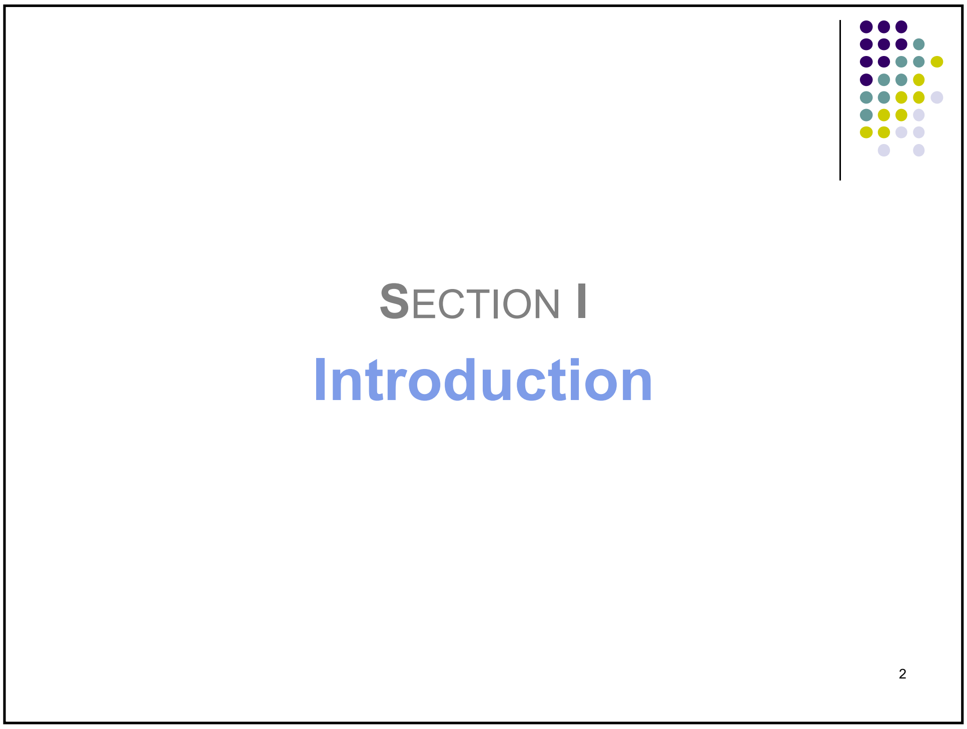
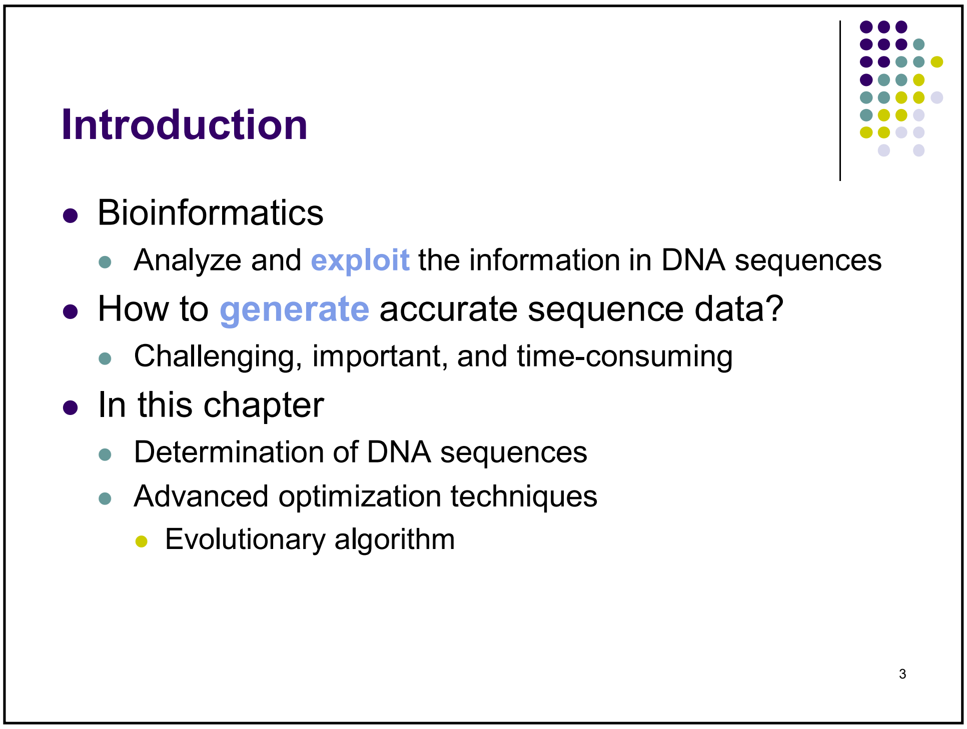


Determining Genome Sequences from Experimental Data Using Evolutionary Computation



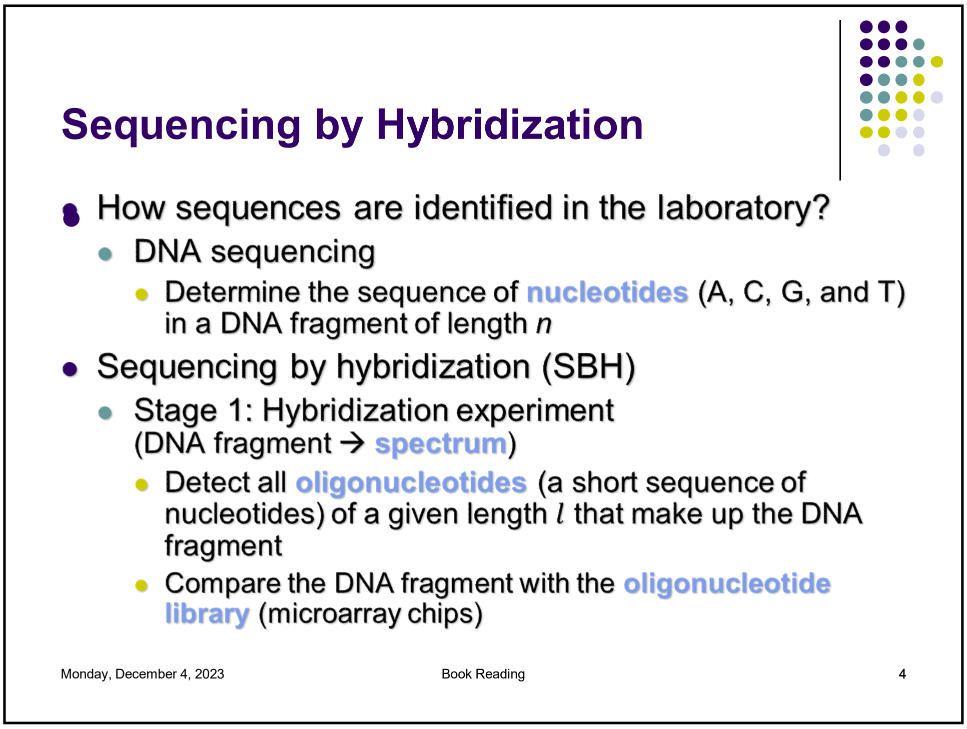
SECTION I

Introduction



Introduction

- Biointformatics
  - Analyze and **exploit** the information in DNA sequences
- How to **generate** accurate sequence data?
  - Challenging, important, and time-consuming
- In this chapter
  - Determination of DNA sequences
  - Advanced optimization techniques
    - Evolutionary algorithm



Sequencing by Hybridization

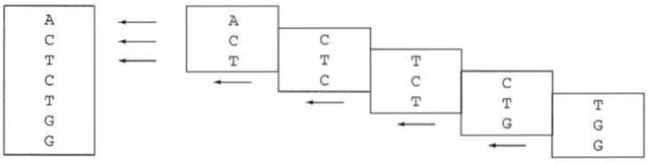
- How sequences are identified in the laboratory?
  - DNA sequencing
    - Determine the sequence of **nucleotides** (A, C, G, and T) in a DNA fragment of length *n*
- Sequencing by hybridization (SBH)
  - Stage 1: Hybridization experiment (DNA fragment → **spectrum**)
    - Detect all **oligonucleotides** (a short sequence of nucleotides) of a given length *l* that make up the DNA fragment
    - Compare the DNA fragment with the **oligonucleotide library** (microarray chips)

Sequencing by Hybridization

- Sequencing by hybridization (SBH)
  - Stage 2: Reconstructing the original sequence (spectrum → sequence)
    - An **ideal** spectrum (Stage 1 is performed **without errors**)
    - Find an ordering of the spectrum elements such that neighboring elements always overlap on l -1 nucleotides

Example: Reconstruction of Sequence from an Ideal Spectrum

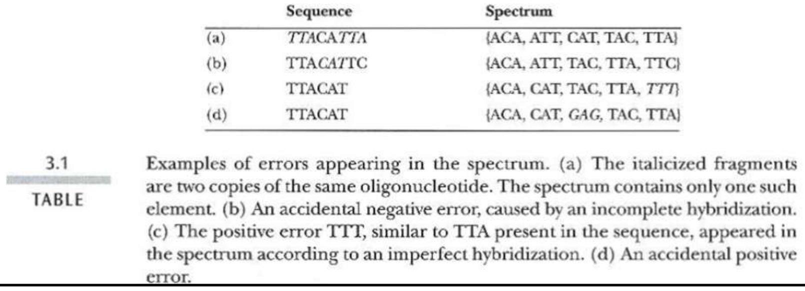
- Original sequence (n=7)
  - ACTCTGG
- Oligonucleotide library (l=3)
  - AAA, AAC, AAG, AAT, ACA, …, TTG, TTT
- Ideal spectrum
  - ACT, CTC, TCT, CTG, TGG
- Reconstructed sequence



3.1 FIGURE The reconstruction of the original sequence from the ideal spectrum. Oligonucleotides (in staggered boxes on the right) are ordered in such a way that neighbors always have $l-1$ nucleotides in common, which leads to the sequence in the box on the left.

Experimental Errors in the Spectrum

- Stage 1 usually produces errors in the spectrum
- Two types of errors
  - Negative errors (a, b)
    - Miss one or more words contained in the original sequence
  - Positive errors (c, d)
    - Contain words that are not found in the original sequence



	Sequence	Spectrum
(a)	*TTA*CA*TTA*	{ACA, ATT, CAT, TAC, TTA}
(b)	TTA*CAT*TC	{ACA, ATT, TAC, TTA, TTC}
(c)	TTACAT	{ACA, CAT, TAC, TTA, *TTT*}
(d)	TTACAT	{ACA, CAT, *GAG*, TAC, TTA}

3.1 TABLE

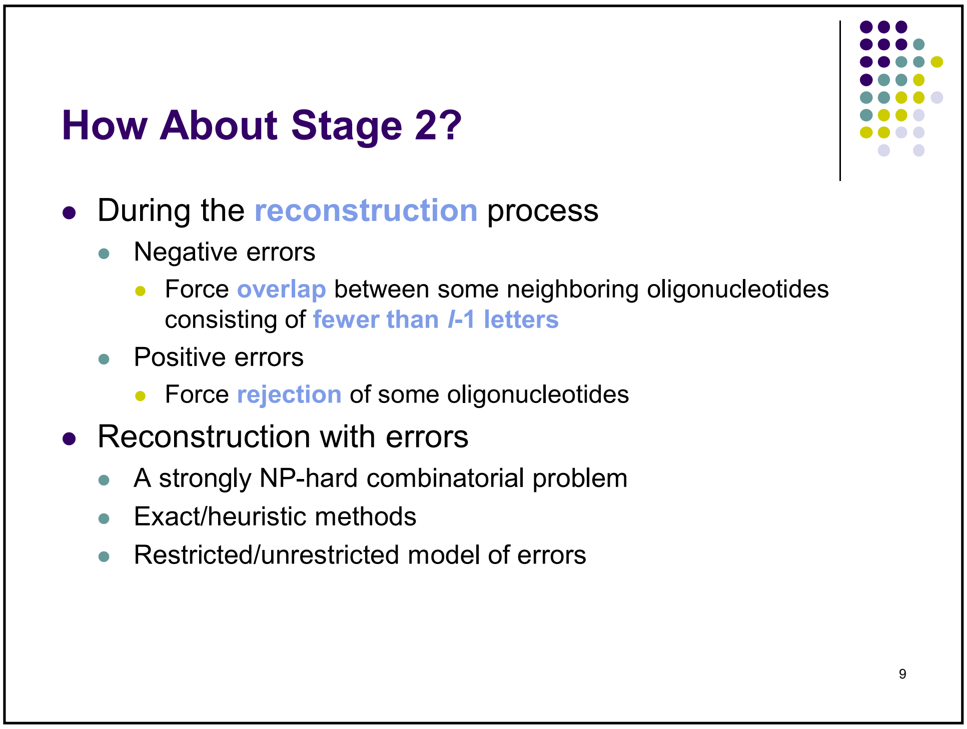
Examples of errors appearing in the spectrum. (a) The italicized fragments are two copies of the same oligonucleotide. The spectrum contains only one such element. (b) An accidental negative error, caused by an incomplete hybridization. (c) The positive error TTT, similar to TTA present in the sequence, appeared in the spectrum according to an imperfect hybridization. (d) An accidental positive error.

7

SECTION II

Formulation of the Sequence Reconstruction Problem

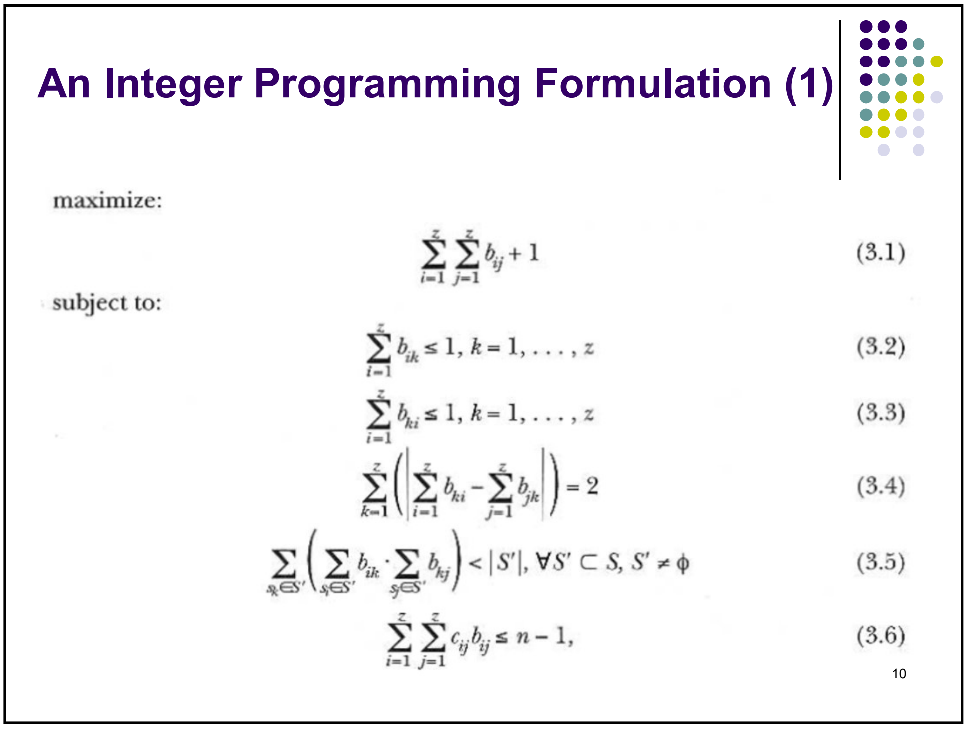
8



How About Stage 2?

- During the **reconstruction** process
  - Negative errors
    - Force **overlap** between some neighboring oligonucleotides consisting of **fewer than *l*-1 letters**
  - Positive errors
    - Force **rejection** of some oligonucleotides
- Reconstruction with errors
  - A strongly NP-hard combinatorial problem
  - Exact/heuristic methods
  - Restricted/unrestricted model of errors

9



An Integer Programming Formulation (1)

maximize:

$$\sum_{i=1}^{z}\sum_{j=1}^{z} b_{ij} + 1 \qquad (3.1)$$

subject to:

$$\sum_{i=1}^{z} b_{ik} \leq 1,\ k = 1, \ldots, z \qquad (3.2)$$

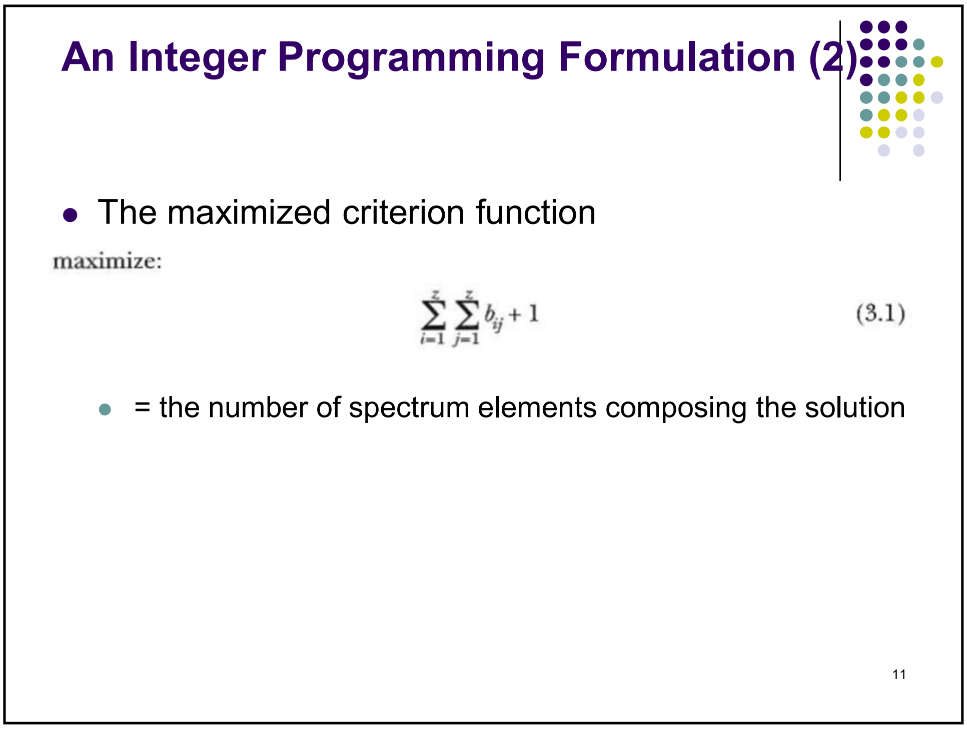
$$\sum_{i=1}^{z} b_{ki} \leq 1,\ k = 1, \ldots, z \qquad (3.3)$$

$$\sum_{k=1}^{z}\left(\left|\sum_{i=1}^{z} b_{ki} - \sum_{j=1}^{z} b_{jk}\right|\right) = 2 \qquad (3.4)$$

$$\sum_{s_k \in S'}\left(\sum_{s_i \in S'} b_{ik} \cdot \sum_{s_j \in S'} b_{kj}\right) < |S'|,\ \forall S' \subset S,\ S' \neq \phi \qquad (3.5)$$

$$\sum_{i=1}^{z}\sum_{j=1}^{z} c_{ij} b_{ij} \leq n - 1, \qquad (3.6)$$

10



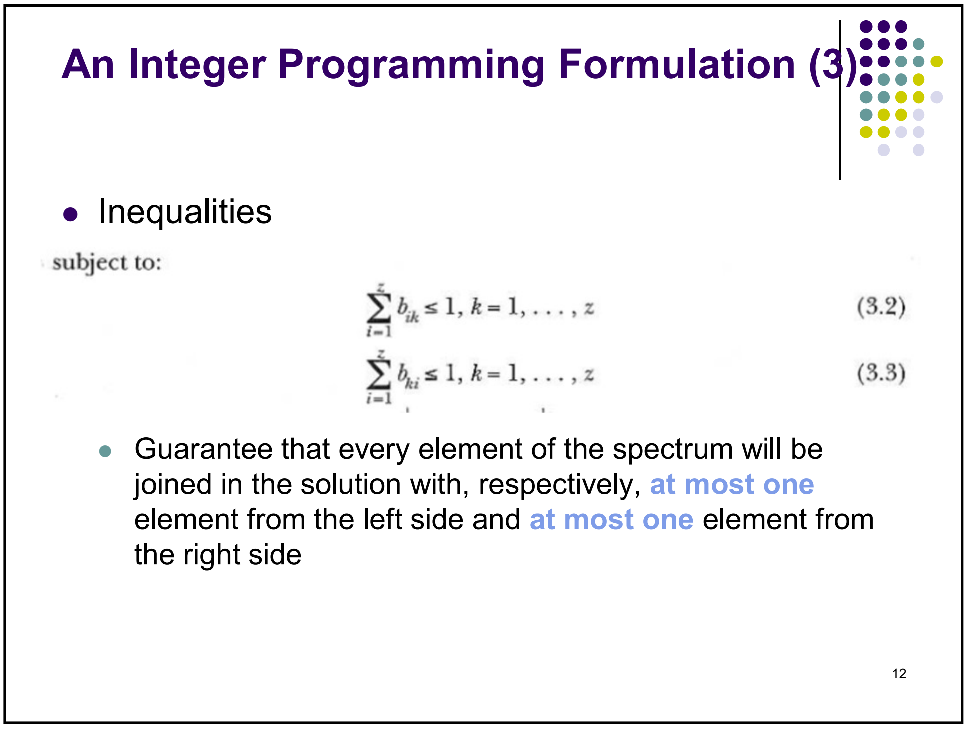
An Integer Programming Formulation (2)

- The maximized criterion function

maximize:

$$\sum_{i=1}^{z}\sum_{j=1}^{z} b_{ij} + 1 \tag{3.1}$$

  - = the number of spectrum elements composing the solution



An Integer Programming Formulation (3)

- Inequalities

subject to:

$$\sum_{i=1}^{z} b_{ik} \le 1,\ k = 1, \ldots, z \tag{3.2}$$

$$\sum_{i=1}^{z} b_{ki} \le 1,\ k = 1, \ldots, z \tag{3.3}$$

  - 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

An Integer Programming Formulation (4)

- Equation

$$\sum_{k=1}^{z}\left(\left|\sum_{i=1}^{z} b_{ki}-\sum_{j=1}^{z} b_{jk}\right|\right)=2 \qquad (3.4)$$

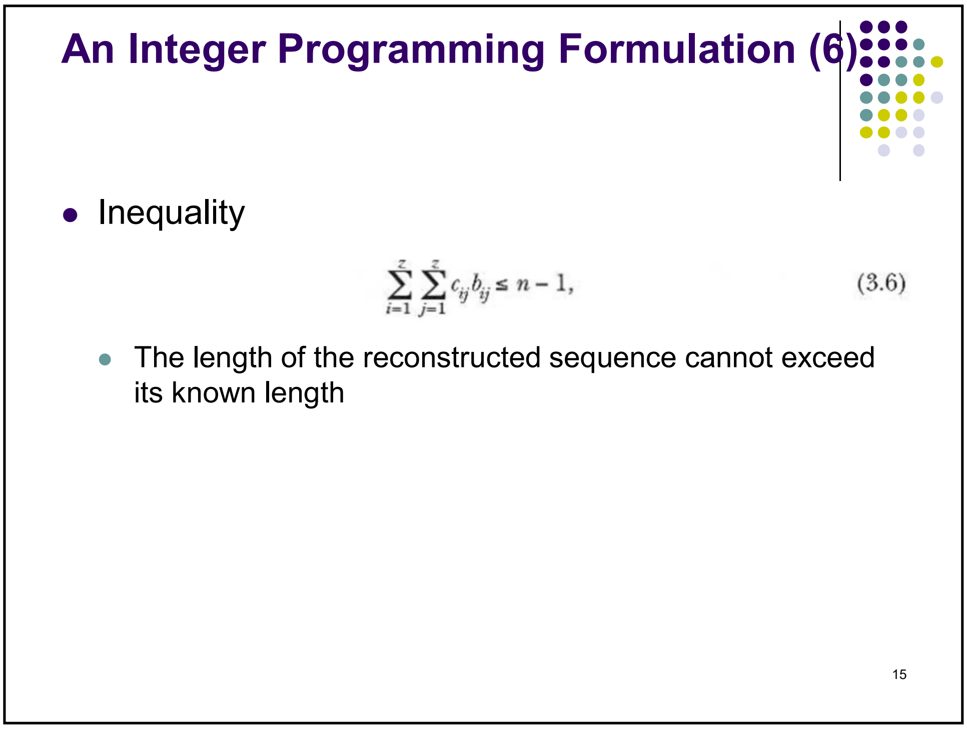
  - Ensure that in any solution, precisely **two elements** appear that are connected to other elements from only one side
    - These elements constitute the **beginning** and the **end** of the reconstructed sequence

An Integer Programming Formulation (5)

- Inequalities

$$\sum_{s_k \in S'}\left(\sum_{s_i \in S'} b_{ik} \cdot \sum_{s_j \in S'} b_{kj}\right)<|S'|, \forall S' \subset S,\, S' \neq \phi \qquad (3.5)$$

  - Allow to eliminate solutions including **subcycles** of elements
    - An element in the solution that is **simultaneously** a **successor** and the **immediate predecessor** of another element in the solution

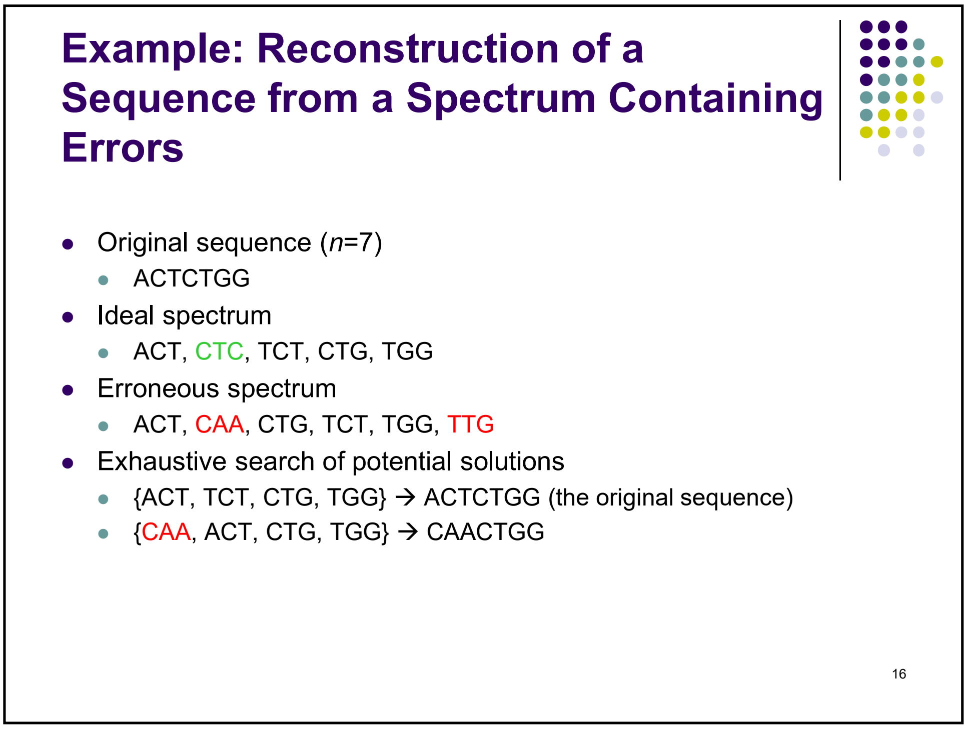


An Integer Programming Formulation (6)

- Inequality

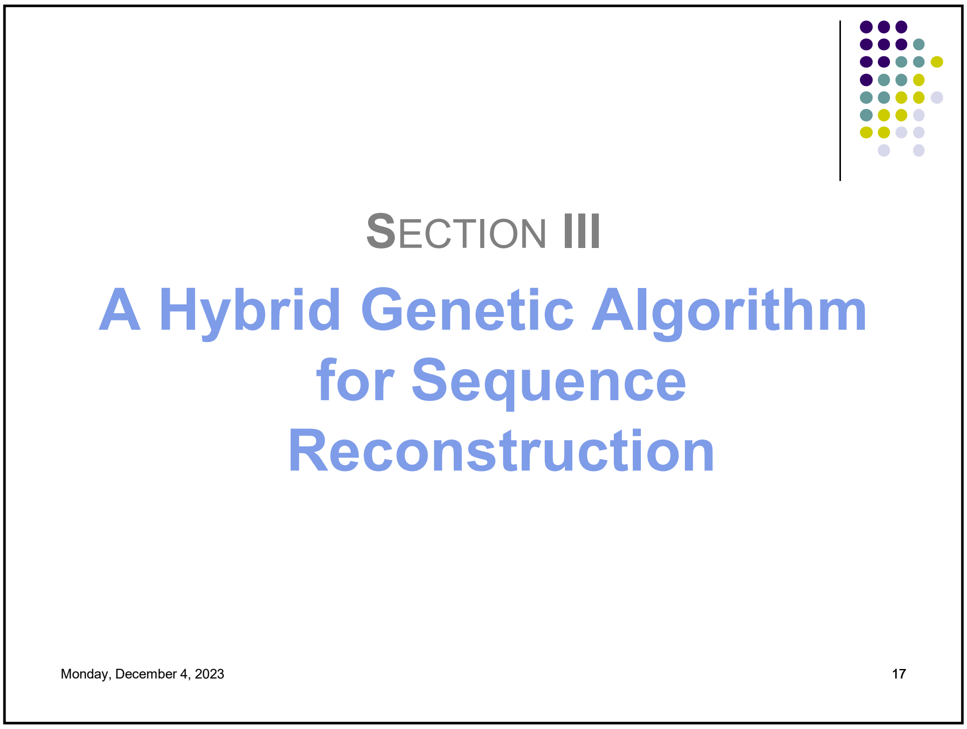
$$\sum_{i=1}^{z}\sum_{j=1}^{z} c_{ij}b_{ij} \le n-1, \qquad (3.6)$$

  - The length of the reconstructed sequence cannot exceed its known length



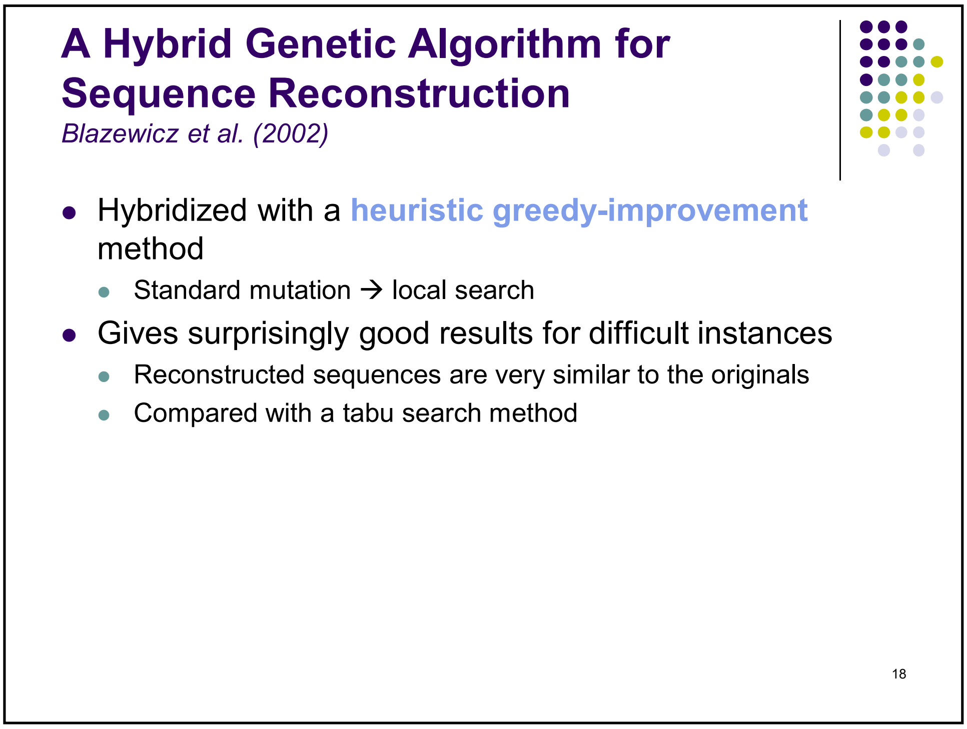
Example: Reconstruction of a Sequence from a Spectrum Containing Errors

- Original sequence (*n*=7)
  - ACTCTGG
- Ideal spectrum
  - ACT, CTC, TCT, CTG, TGG
- Erroneous spectrum
  - ACT, CAA, CTG, TCT, TGG, TTG
- Exhaustive search of potential solutions
  - {ACT, TCT, CTG, TGG} → ACTCTGG (the original sequence)
  - {CAA, ACT, CTG, TGG} → CAACTGG



SECTION III

A Hybrid Genetic Algorithm for Sequence Reconstruction



A Hybrid Genetic Algorithm for Sequence Reconstruction

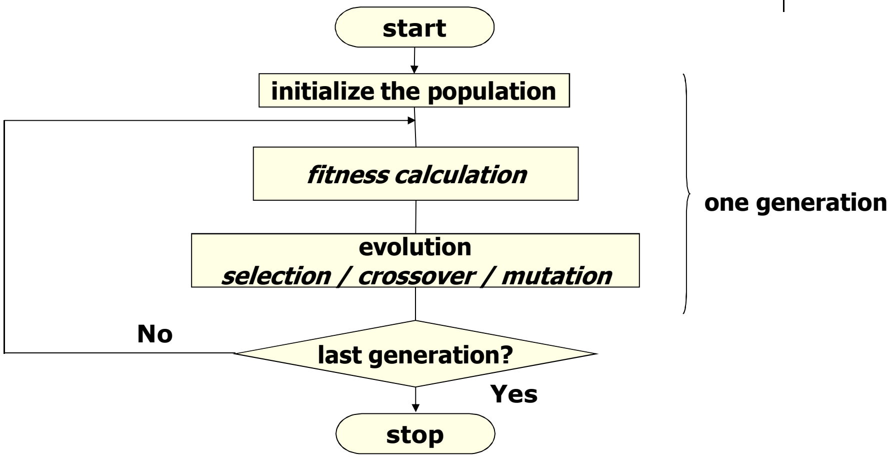
Blazewicz et al. (2002)

- Hybridized with a **heuristic greedy-improvement** method
  - Standard mutation → local search
- Gives surprisingly good results for difficult instances
  - Reconstructed sequences are very similar to the originals
  - Compared with a tabu search method

Criterion Function and Constraint

- Given
  - A spectrum of elements (all of length l)
- To find
  - A solution that **maximizes the number of elements**
    - An ordering of elements chosen from the spectrum, with a corresponding reconstructed sequence
  - The **maximum sequence length** *n*
    - The reconstructed sequence should not be longer than *n*
  - The **general model of errors**
    - Nothing is assumed about the types and numbers of the errors that may exist in the spectrum

Flowchart of Genetic Algorithm



start

initialize the population

fitness calculation

evolution
selection / crossover / mutation

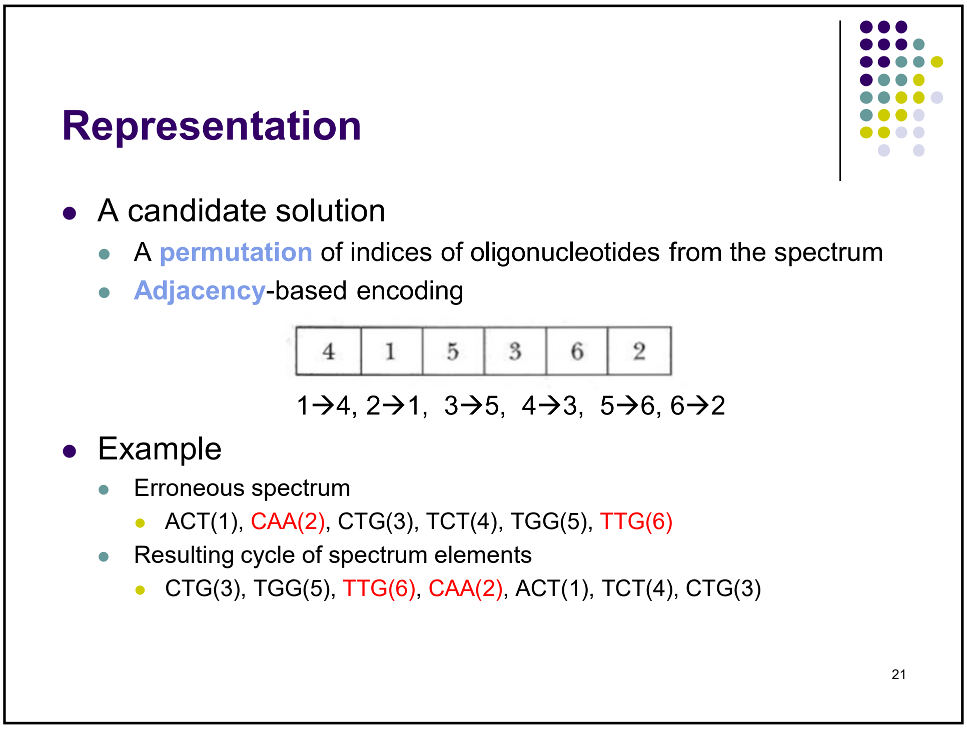
last generation?

No

Yes

stop

one generation



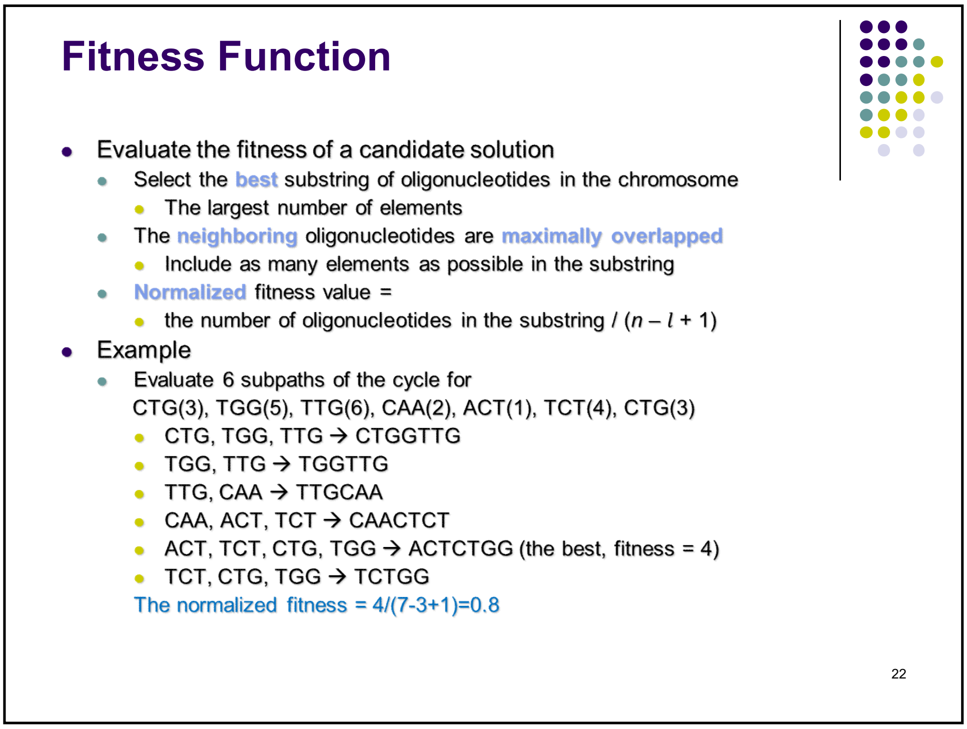
Representation

- A candidate solution
  - A **permutation** of indices of oligonucleotides from the spectrum
  - **Adjacency**-based encoding

4	1	5	3	6	2

1→4, 2→1, 3→5, 4→3, 5→6, 6→2

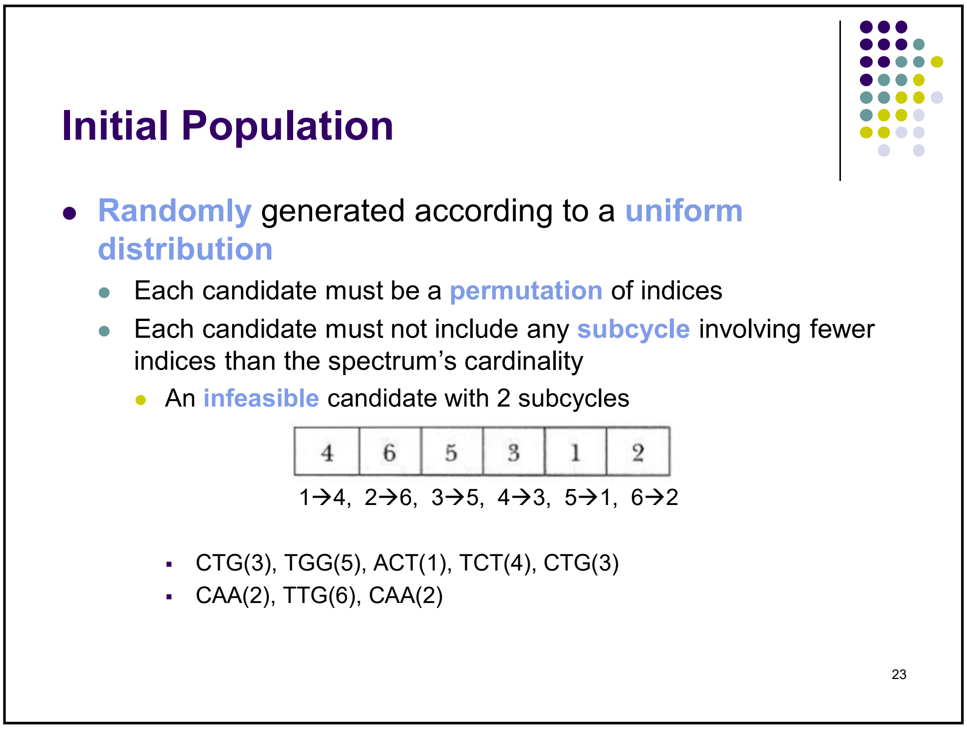
- Example
  - Erroneous spectrum
    - ACT(1), CAA(2), CTG(3), TCT(4), TGG(5), TTG(6)
  - Resulting cycle of spectrum elements
    - CTG(3), TGG(5), TTG(6), CAA(2), ACT(1), TCT(4), CTG(3)



Fitness Function

- Evaluate the fitness of a candidate solution
  - Select the **best** substring of oligonucleotides in the chromosome
    - The largest number of elements
  - The **neighboring** oligonucleotides are **maximally overlapped**
    - Include as many elements as possible in the substring
  - **Normalized** fitness value =
    - the number of oligonucleotides in the substring / ($n - l + 1$)
- Example
  - Evaluate 6 subpaths of the cycle for
    CTG(3), TGG(5), TTG(6), CAA(2), ACT(1), TCT(4), CTG(3)
    - CTG, TGG, TTG → CTGGTTG
    - TGG, TTG → TGGTTG
    - TTG, CAA → TTGCAA
    - CAA, ACT, TCT → CAACTCT
    - ACT, TCT, CTG, TGG → ACTCTGG (the best, fitness = 4)
    - TCT, CTG, TGG → TCTGG

    The normalized fitness = 4/(7-3+1)=0.8



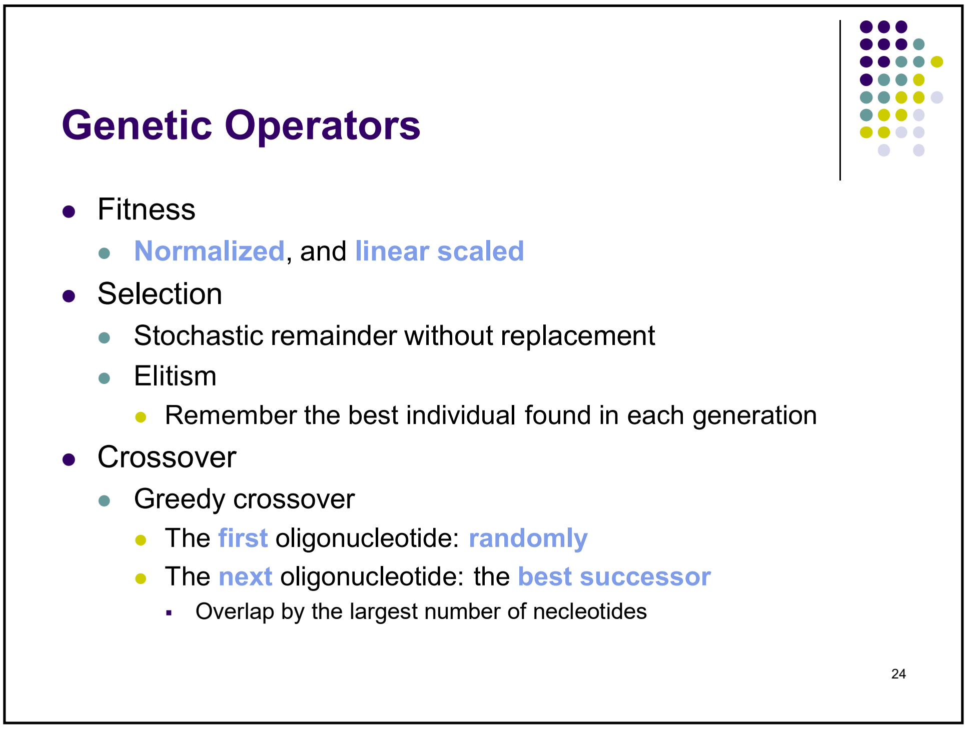
Initial Population

- Randomly generated according to a uniform distribution
  - Each candidate must be a permutation of indices
  - Each candidate must not include any subcycle involving fewer indices than the spectrum's cardinality
    - An infeasible candidate with 2 subcycles

4	6	5	3	1	2

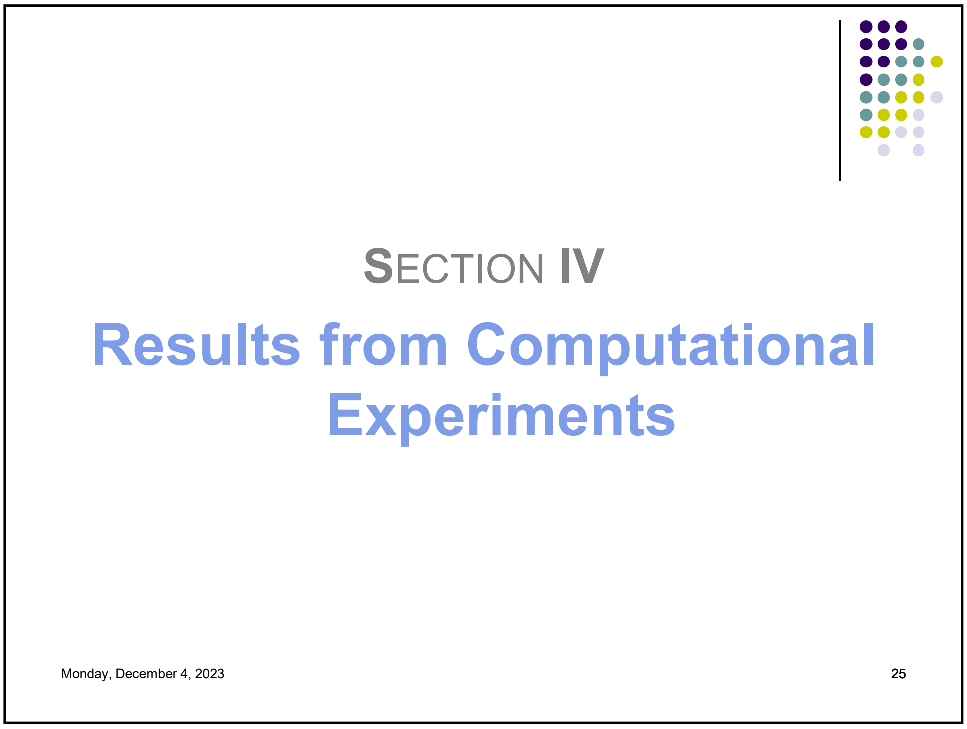
1→4, 2→6, 3→5, 4→3, 5→1, 6→2

- CTG(3), TGG(5), ACT(1), TCT(4), CTG(3)
- CAA(2), TTG(6), CAA(2)



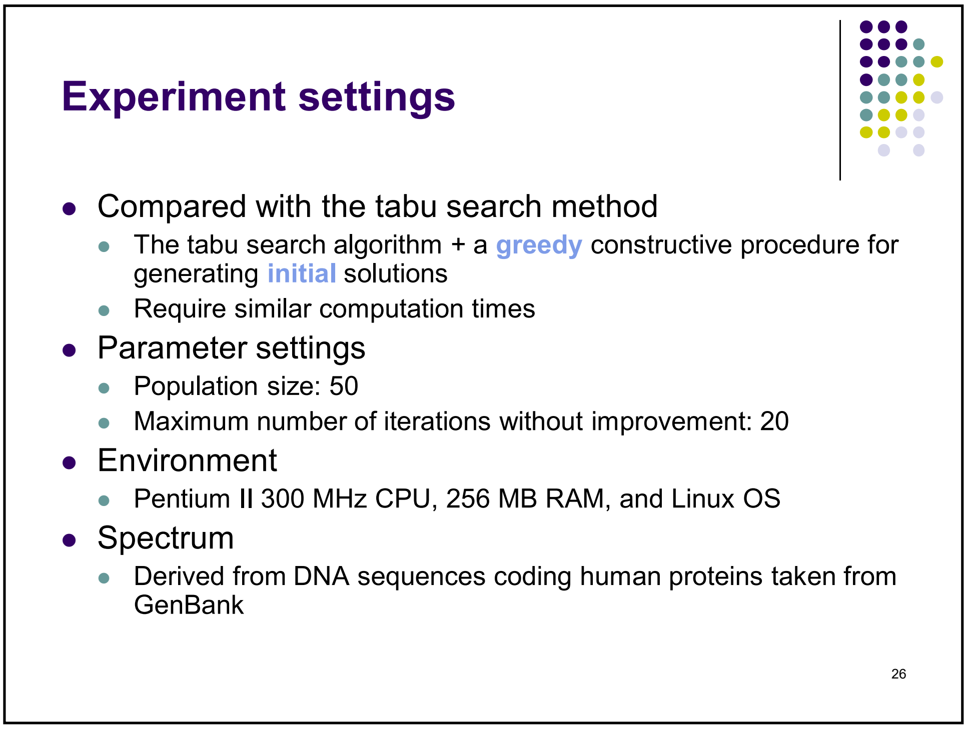
Genetic Operators

- Fitness
  - Normalized, and linear scaled
- Selection
  - Stochastic remainder without replacement
  - Elitism
    - Remember the best individual found in each generation
- Crossover
  - Greedy crossover
    - The first oligonucleotide: randomly
    - The next oligonucleotide: the best successor
      - Overlap by the largest number of necleotides



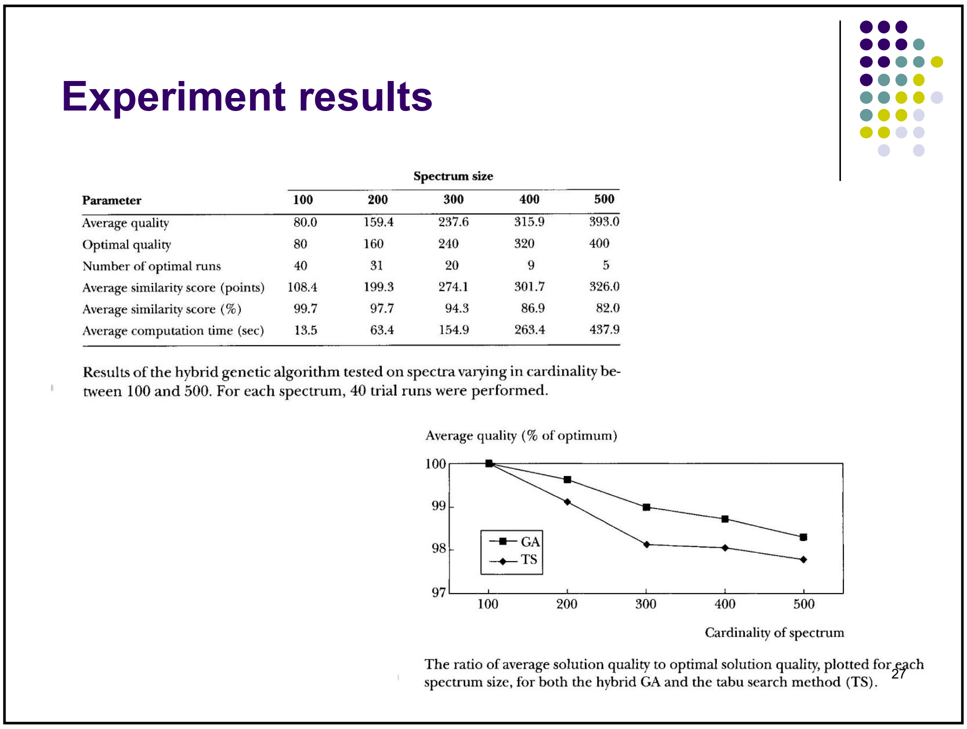
SECTION IV

Results from Computational Experiments



Experiment settings

- Compared with the tabu search method
  - The tabu search algorithm + a **greedy** constructive procedure for generating **initial** solutions
  - Require similar computation times
- Parameter settings
  - Population size: 50
  - Maximum number of iterations without improvement: 20
- Environment
  - Pentium II 300 MHz CPU, 256 MB RAM, and Linux OS
- Spectrum
  - Derived from DNA sequences coding human proteins taken from GenBank



Experiment results

	Spectrum size				
Parameter	**100**	**200**	**300**	**400**	**500**
Average quality	80.0	159.4	237.6	315.9	393.0
Optimal quality	80	160	240	320	400
Number of optimal runs	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

Results of the hybrid genetic algorithm tested on spectra varying in cardinality between 100 and 500. For each spectrum, 40 trial runs were performed.

The ratio of average solution quality to optimal solution quality, plotted for each spectrum size, for both the hybrid GA and the tabu search method (TS).

The End