Protein similarity search under mRNA structural constraints: application to selenocysteine incorporation

Rolf Backofen* N.S. Narayanaswamy† Firas Swidan
Oettingenstrasse 67, 80538 Munich, Germany
Tel: +49 89 2178 2213 Fax: +49 89 2178 2238
Email: {backofen,swamy,swidan}@informatik.uni-muenchen.de

Abstract

Selenocysteine is the 21th amino acid, which occurs in all kingdoms of life. Selenocysteine is encoded by the STOP-codon UGA. For its insertion, it requires a specific mRNA sequence downstream the UGA-codon that forms a hairpin like structure (called Sec insertion sequence (SECIS)). We consider the computational problem of generating new amino acid sequences containing selenocysteine. This requires to find an mRNA sequence that is similar to the SECIS-consensus, is able to form the secondary structure required for selenocysteine insertion, and whose translation is maximally similar to the original amino acid sequence. We show that the problem can be solved in linear time when the structure does not contain pseudoknots.
Keywords: Selenocysteine, SECIS, Protein Engineering

1 Introduction

Selenocysteine (Sec) is a rare amino acid, which was discovered as the 21st amino acid [1] about a decade ago. Proteins containing selenocysteine are consequently called *selenoproteins*. The discovery of selenocysteine was another clue to the complexity and flexibility of the mRNA translation mechanism. Selenocysteine is encoded by the UGA-codon, which is usually a STOP-codon encoding the end of translation by the ribosome. It has been shown [1] that in the case of selenocysteine, termination of translation is inhibited in the presence of a specific mRNA sequence in the 3'-region after the UGA-codon that forms a hairpin like structure (called Sec insertion sequence (SECIS), see Figure 1).

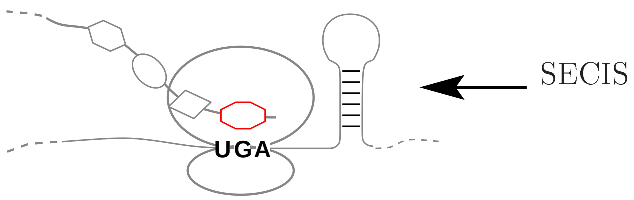

Figure 1: Translation of mRNA requires a SECIS-element in case of selenocysteine.

*Partially supported by the DFG within the national program SPP 1087 "Selenoprotein – Biochemische Grundlagen und klinische Bedeutung"

†Supported by DFG Grant No. Jo 291/2-1

Concerning bioinformatics approaches in the area of selenoproteins, we have the following situation. An obvious problem is to search for existing but unknown selenoproteins. However, we consider a different problem, namely modifying existing proteins such that selenocysteine is incorporated instead of another amino acid (e.g., instead of a cysteine). There are two possible reasons for wanting such modifications. First, it provides a possibility of generating new, functionally modified selenoproteins. Experimentally, this has been done successfully for *E. coli* in [5], where the mRNA sequence of a protein not containing selenocysteine was engineered (by hand) such that it forms a *E. coli* SECIS (while preserving maximum similarity on the amino acid level). The engineered protein now had a selenocysteine in place of a catalytic cysteine (at position 41), and showed a 4-fold enhanced catalytic activity. Second, and more important, seleno-variants of proteins could be used for the phase determination in an X-ray crystallography (or in the NMR-analysis) of a given protein.

In this paper, we consider the computational properties of substituting one position by a selenocysteine in a given amino acid sequence. Let $S = S_1 \dots S_{3n}$ be the consensus of the SECIS-element, and let $A = A_1 \dots A_n$ be the original amino acid sequence following the position where we wish to insert selenocysteine. By modifying an mRNA sequence to insert a selenocysteine the sequence A may be modified. We have to find an appropriate mRNA sequence $N = N_1 \dots\dots N_{3n}$, which is a SECIS element and encodes an amino acid sequence $A' = A'_1 \dots A'_n$ that has maximum similarity with A. Thus, we have the following picture:

$$\begin{array}{lccccc}
A = & A_1 & \dots & A_i & \dots & A_n \\
 & \wr & & \wr & & \wr \\
A' = & A'_1 & \dots & A'_i & \dots & A'_n \\
 & \uparrow & & \uparrow & & \uparrow \\
N = & \overbrace{N_1 N_2 N_3} & \dots & \overbrace{N_{3i-2}N_{3i-1}N_{3i}} & \dots & \overbrace{N_{3n-2}N_{3n-1}N_{3n}} \\
 & \wr\ \ \wr\ \ \wr & & \wr\ \ \wr\ \ \wr & & \wr\ \ \wr\ \ \wr \\
S = & S_1\ S_2\ S_3 & \dots & S_{3i-2}S_{3i-1}S_{3i} & \dots & S_{3n-2}S_{3n-1}S_{3n}.
\end{array}$$

The similarities on both the amino acid ($A_i \sim A'_i$) and nucleotide level ($N_j \sim S_j$) will be measured by functions $F_{A_i}^{S_{3i-2}S_{3i-1}S_{3i}}(N_{3i-2}N_{3i-1}N_{3i})$, and we search an N that maximizes $\sum_{i=1}^{n} F_{A_i}^{S_{3i-2}S_{3i-1}S_{3i}}(N_{3i-2}N_{3i-1}N_{3i})$, (referred to as *cost* or *similarity*) and satisfies the constraints given by the SECIS structure. These constraints are given by the bonds realized in the structure. If there is a bond between position r and s in the SECIS, then every valid N has to satisfy that N_r and N_s are complementary.

Using the fact that the structure of the SECIS element is a hairpin, we show that the described problem can be solved in linear time. The property that makes the problem computationally easy is that in the linear embedding of the hairpin structure, no edges cross. We have shown that it is NP-hard to solve the problem optimally if every linear embedding has crossing edges (to be presented in the complete version of this paper). This case can occur if we have pseudoknots as an underlying structure. Pseudoknots are important for programmed frameshifts, which allow to encode two different amino acid sequences in one mRNA sequence. This mechanism has been identified in viruses as well as in prokaryotes and eukaryotes [2, 3].

2 Preliminaries

We follow graph theoretic notations and definitions as presented in the standard text by Harary [4]. A labeled graph $\Gamma(V, E, \mathrm{Lab})$ is an undirected graph $\Gamma(V, E)$ along with a function

Lab : $E \to X$, where X is a finite set whose elements are called *edge labels*. In this paper, $X = \{1, -1\}$. An undirected graph is said to be *outer-planar* if it can be drawn satisfying the following conditions: 1.) all the vertices lie on a line, 2) all the edges lie on one side of the line, and 3.) two edges that intersect in the drawing do so only at their end points. Such a drawing is called an outer-planar embedding of the graph.
The solution to the problem we address in this paper is a string whose letters are from the set $\{A, C, G, U\}$. The elements of this set are referred to as *nucleotides* and an element of $\{A, C, G, U\}^3$ is referred to as a *codon*. Biologically, A and U are complements of each other and so are G and C. The complement of a variable, taking nucleotide values, is denoted by the superscript $\complement$. Now our problem can be stated as follows:

Input: An edge labeled graph $\Gamma = (V, E, \mathrm{Lab})$ on $3n$ vertices, $V = \{v_1, \dots, v_{3n}\}$. For every edge $\{v_k, v_l\} \in E$, the label $\mathrm{Lab}(v_k, v_l)$ is either 1 or -1. If $\mathrm{Lab}(v_k, v_l) = +1$, then we have a bond between the nucleotides at positions k and l in the mRNA. Otherwise, there must not be a bond between positions k and l. n functions, $f_1, \dots, f_n$ are also part of the input. f_i is associated with $\{v_{3i-2}, v_{3i-1}, v_{3i}\}, 1 \le i \le n$, and is a function defined on $\{A, C, G, U\}^3$ and takes rational values.

Output: Find vector $(N_1, \dots, N_{3n}) \in \{A, C, G, U\}^{3n}$ s.t. N_k is assigned to v_k and

1. $\{v_k, v_l\} \in E(\Gamma)$ and $\mathrm{Lab}(v_k, v_l) = 1$ implies that $N_k = N_j^\complement$.
2. $\{v_k, v_l\} \in E(\Gamma)$ and $\mathrm{Lab}(v_k, v_l) = -1$ implies that $N_k \ne N_j^\complement$.
3. $\sum_{i=1}^n f_i(N_{3i-2}, N_{3i-1}, N_{3i})$ is maximized. This function is referred to as the *similarity* or *cost* of the assignment at which it is evaluated.

We denote the above problem by $MRSO(\Gamma, f_1, \dots, f_n)$. In the following, we will refer to conditions 1 and 2 together as the *complementarity condition*. We will call edges that are labeled with 1 *bonds*, and edges labeled by -1 *prohibited bonds*. Prohibited bonds are necessary since the SECIS also needs some bulged nucleotides.

About Graphs: Γ is referred to as the *structure graph*. A structure graph that specifies only bonds is called an *RNA-graph*. For RNA-graphs, the degree of every node is at most one. Given a structure graph Γ, we define the *implied graph* Γ^{impl} to be a graph on the vertices $\mathrm{V}(\Gamma^{\mathrm{impl}}) = \{u_1, \dots, u_n\}$ with

$$\mathrm{E}(\Gamma^{\mathrm{impl}}) = \left\{ \{u_i, u_j\} \;\middle|\; \begin{array}{l} \exists r \in \{3i-2, 3i-1, 3i\} : \\ \quad \exists s \in \{3j-2, 3j-1, 3j\} : (v_r, v_s) \in E(\Gamma) \end{array} \right\}.$$

We follow the convention that, independent of the subscript or superscript, u denotes a vertex from $\mathrm{V}(\Gamma^{\mathrm{impl}})$ and v denotes a vertex from Γ. Given $I \subseteq \{1..n\}$, U_I denotes $\{u_j \mid j \in I\}$. $\mathrm{E}(\Gamma^{\mathrm{impl}})|_I$ denotes the edge set of the induced subgraph of Γ^{impl} on U_I. $\mathrm{E}(\Gamma)||_I$ denotes the edge set of the induced subgraph of Γ on $\{v_{3i-2}, v_{3i-1}, v_{3i} \mid i \in I\}$.

About Codons: Given a sequence of codons $L_1 \dots L_n$, let $N_1 \dots N_{3n}$ be the corresponding nucleotide sequence obtained by replacing L_i by the corresponding nucleotide representation $N_{3i-2}N_{3i-1}N_{3i}$. $L_1 \dots L_n$ is said to *satisfy* $E(\Gamma)$ iff the corresponding nucleotide representation $\{N_{3i-2}N_{3i-1}N_{3i}\}_{i=1}^n$ satisfies the complementarity conditions for Γ. L_i and L_j are said to be *valid for* $\{u_i, u_j\}$ *w.r.t.* $E(\Gamma)$ if the corresponding nucleotides $N_{3i-2}N_{3i-1}N_{3i}$ and $N_{3j-2}N_{3j-1}N_{3j}$ satisfies the complementarity condition imposed by $E(\Gamma)$.

3 Algorithm in the absence of pseudoknots

We present a polynomial time dynamic programming algorithm to solve $MRSO$ when the Γ^{impl} is outerplanar. The algorithm is based on a recurrence relation that we prove in this section. Also, the algorithm solves $MRSO$ even when the functions can take arbitrary values. In the presentation below, we assume that Γ^{impl} is an outerplanar graph and $\{u_1, \ldots, u_n\}$ is the ordering of the vertices from left to right on the line in an outer-planar embedding. Furthermore, the function associated with u_i is f_i for $1 \leq i \leq n$. Having fixed an embedding of the graph on the line, we do not distinguish between a vertex and its index. That is, the interval $[i, \ldots, i+k]$ denotes the set of vertices $\{u_i, \ldots, u_{i+k}\}$.
We define the relation $(i, L_i) \equiv_{E(\Gamma)} (j, L_j)$, where $(i, L_i) \equiv_{E(\Gamma)} (j, L_j)$ is true if and only if either $\{u_i, u_j\} \notin \mathrm{E}(\Gamma^{\text{impl}})$, or L_i and L_j are *valid* for $\{u_i, u_j\}$ w.r.t. $E(\Gamma)$. We now define the following function:

$$w(i, i+k, L_i, L_{i+k}) = \max_{L_{i+1},\ldots,L_{i+k-1}} \left\{ \sum_{i \leq j \leq i+k} f_j(L_j) \mid L_i, \ldots, L_{i+k} \text{ satisfies } \mathrm{E}(\Gamma)|_{[i..i+k]} \right\}.$$

If the above set is empty, the value of the function is considered as $-\infty$. This function forms the central part of our algorithm to solve the problem because $\max_{L_1, L_n} w(1, n, L_1, L_n)$ is the value that we are interested in computing. The base cases for w is the following:

$$w(i, i+1, L_i, L_{i+1}) = \begin{cases} f_i(L_i) + f_{i+1}(L_{i+1}) & \text{if } (i, L_i) \equiv_{E(\Gamma)} (i+1, L_{i+1}) \\ -\infty & \text{otherwise.} \end{cases}$$

We implement the recurrence relation efficiently by splitting an interval $[i..i+k]$ into two parts and solving the resulting subproblems. If there is no edge departing from i to vertex in the interval $[i..i+k-1]$, the interval can just be split into $[i, i+1]$ and $[i+1, i+k]$. If there is such an edge, we choose the farthest vertex p in $[i..i+k-1]$ which is adjacent to i. The edge (i, p) is called the *maximal* edge in $\mathrm{E}(\Gamma^{\text{impl}})|_{[i..i+k-1]}$. Then, the interval is split into $[i, p]$ and $[p, i+k]$. Hence, define $\mathrm{next}(i, i+k))$ for $k \geq 2$ by

$$\mathrm{next}(i, i+k) = \begin{cases} i+r & \text{if } \{i, i+r\} \text{ is maximal in } \mathrm{E}(\Gamma^{\text{impl}})|_{[i..i+k-1]} \\ i+1 & \text{otherwise.} \end{cases}$$

Theorem 3.1 *Let* $(\Gamma, f_1, \ldots, f_n)$ *be instance of the* $MRSO$*. For* $k \geq 2, 1 \leq i \leq n-k$,

$$w(i, i+k, L_i, L_{i+k}) = \begin{cases} -\infty & (i, L_i) \not\equiv_{E(\Gamma)} (i+k, L_{i+k}) \\ \max_{L_p} \begin{pmatrix} & w(i, p, L_i, L_p) \\ + & w(p, i+k, L_p, L_{i+k}) \\ - & f_p(L_p) \end{pmatrix} & \textit{if } (i, L_i) \equiv_{E(\Gamma)} (i+k, L_{i+k}), \end{cases} \tag{1}$$

where $p = \mathrm{next}(i, i+k)$.

The proof is given in the appendix.

This theorem gives us an efficient way of finding the optimum value of $MRSO$ for an input $(\Gamma, f_1, \ldots, f_n)$. We define an additional trace matrix T so as to obtain an assignment of codons at which the optimum is attained. Define $T(i, i+k, L_i, L_{i+k})$ for $k \geq 2$ recursively as follows:

$$T(i, i+k, L_i, L_{i+k}) = \begin{cases} \text{undef} & \text{if } (i, L_i) \not\equiv_{E(\Gamma)} (i+k, L_{i+k}) \\ (p, (L_p^1, \ldots, L_p^r)) & \text{otherwise,} \end{cases} \tag{2}$$

where $p = \text{next}(i, i+k)$ and $(L_p^1, \ldots, L_p^r)$ is the list of all codons such that for all $t \in \{1, \ldots, r\}$, we have $w(i, p, L_i, L_p^t) + w(p, i+k, L_p^t, L_{i+k}) - f_p(L_p^t) = w(i, i+k, L_i, L_{i+k})$. By the recurrence relations given for the w and T matrices, it is clear that these matrices can easily be calculated using a simple dynamic programming algorithm. This algorithm will have an $O(n^2)$ runtime. The algorithm can be implemented in linear time for RNA-graphs (or general graphs with bounded degree as in the case of SECIS-elements) by observing that not all matrix elements (corresponding to interval of the codon sequence positions) are considered, but only those that are generated by the splitting induced by the $\text{next}(i, i+k)$ function.

Therefore, for calculating $w(1, n, L_1, L_n)$, we need not all entries of the matrix w but only the entries of the forms $w(1, next(1, n), \cdot, \cdot)$, $w(1, next(1, next(1, n), \cdot, \cdot)) \ldots w(next(1, n), n, \cdot, \cdot)$. Thus, only $64 \times 64 \times |E|$ many entries are visited. This is captured in a binary tree-like representation of those entries, and a recursive algorithm to fill these entries.

4 Acknowledgment

The first author likes to thank Prof. Böck from the Institute of Microbiology for explaining him the biochemical background of selenocysteine, for many suggestion and the fruitful cooperation on the problems discussed in this paper. Furthermore, he likes to thank Prof. Clote for pointing out the problem, and for initiating the collaboration with Prof. Böck. He would also like to thank Sebastian Will for many discussion, and for reading draft versions of this paper. The authors thank the anonymous referees for their comments.

References

[1] A. Böck, K. Forchhammer, J. Heider, and C. Baron. Selenoprotein synthesis: an expansion of the genetic code. *Trends Biochem Sci*, 16(12):463–467, 1991.

[2] P. J. Farabaugh. Programmed translational frameshifting. *Microbiology and Molecular Biology Reviews*, 60(1):103–134, 1996.

[3] D. P. Giedroc, C. A. Theimer, and P. L. Nixon. Structure, stability and function of rna pseudoknots involved in stimulating ribosomal frameshifting. *Journal of Molecular Biology*, 298(2):167–186, 2000.

[4] Frank Harary. Graph theory. *Addison Wesley*, 1972.

[5] Stephane Hazebrouck, Luc Camoin, Zehava Faltin, Arthur Donny Strosberg, and Yuval Eshdat. Substituting selenocysteine for catalytic cysteine 41 enhances enzymatic activity of plant phospholipid hydroperoxide glutathione peroxidase expressed in *escherichia coli*. *Journal of Biological Chemistry*, 275(37):28715–28721, 2000.

[6] Gert Smolka. The Oz programming model. In Jan van Leeuwen, editor, *Computer Science Today*, Lecture Notes in Computer Science, vol. 1000, pages 324–343. Springer-Verlag, Berlin, 1995.

A Appendix

Results We have implemented the program in the constraint programming language Oz [6]. We have then used our program to search for proteins allowing selenocysteine incorporation in the complete genome of *E. coli*. Using our program, we have found modifications of several proteins that have high similarities to both the original amino-acid sequence, and to the SECIS-element, and allow to form the hairpin structure as required. These results are shown below in Table 1.

name	ind	PAM	orgAmino	modAmino	mRNA
yahE	88	53	RRWLSPTLQM	RRWLVPSLDL	CGACGAUGGUUGGUACCAAGUCUGGACCUA
mhpC	2	56	RIWLVKRQNR	RIWLVRRRDR	CGAAUAUGGUUGGUACGCAGGCGCGACCGA
ybdJ	55	53	LWFLVLGAIE	LWFLVPGPVE	CUAUGGUUCCUCGUACCGGGUCCGGUCGAG
ybiW	123	54	PWWRGQTVQD	PWWRVRSLDE	CCAUGGUGGCGCGUACGAAGUCUCGACGAG
ycaI	371	54	PEWQLPPVLR	PEWQLPSLVR	CCAGAAUGGCAGCUACCAAGUCUGGUGCGG
hyaF	114	54	GLWRVRRRRG	PLWRVRRRDG	CCAUUAUGGCGCGUACGCAGGCGCGACGGG
holB	158	53	RLHYLAPPPE	RLHYLASPPE	CGAUUACACUACCUAGCGAGUCCGCCGGAA
ymfO	141	53	QRWPEGDRRE	QRWPVGGRRE	CAACGAUGGCCCGUAGGCGGUCGCCGCGAG
hnr	320	59	QIWGTGGRLR	QIWGVGGRLR	CAAAUAUGGGGGGUAGGCGGUCGCCUCCGC
yciS	59	53	GLFWLRVRVS	PLFWLRSRVP	CCAUUAUUCUGGCUACGCAGUCGCGUGCCA
yeeP	105	53	HEWDMAGIQP	HEWELAGLQP	CACGAAUGGGAGCUAGCAGGUCUGCAGCCC
b2085	10	53	RWYLMGEGEM	RWYLLGSPQL	CGAUGGUACUUGCUAGGGAGUCCCCAGCUA
b2710	430	53	QWIYDPAKGE	QWIYVPSRGE	CAAUGGAUAUACGUACCCAGUCGGGGCGAA
ygcX	171	54	DWYRLRHEEA	QWYRLRRREA	CAAUGGUACCGCCUACGCAGGCGCGAGGCG
recC	243	53	RYYWGDIKDP	RYYWVPSRDP	CGAUACUACUGGGUACCCAGUCGGGACCCA
b3027	96	57	GWWLFWGRFI	PWWLLRGRVV	CCAUGGUGGCUCCUACGCGGUCGCGUGGUG
yhcR	79	54	LFYLISRLFV	LFYLVARLLV	CUAUUCUACCUCGUAGCAAGGCUGCUCGUG
yrfG	83	53	LDYWSEQLGL	LDYWVPRLGL	CUAGACUACUGGGUACCAAGGCUGGGCCUA
yidL	221	53	SEAWLRRLFL	PEAWLRRLVL	CCAGAAGCAUGGCUACGAAGGCUCGUGCUA
yieN	290	54	LWYDAQSLNL	LWYEVRSLDL	CUAUGGUACGAGGUACGAAGUCUCGACCUC
yjiM	99	53	PYFYFSDLVV	PYFYLPGLVV	CCAUACUUCUACCUACCAGGUCUGGUGGUA

Table 1: Some Results for *E. coli* genome

Proof of Theorem 3.1. For the case when $(i, L_i) \not\equiv_{E(\Gamma)} (i+k, L_{i+k})$, the result is true because we follow the convention that the maximum over an empty set is $-\infty$. So we consider the case when $(i, L_i) \equiv_{E(\Gamma)} (i+k, L_{i+k})$. So let $k \geq 2$. We wish to evaluate,

$$\max_{L_p} \begin{pmatrix} & w(i,p,L_i,L_p) \\ + & w(p,i+k,L_p,L_{i+k}) \\ - & f_p(L_p). \end{pmatrix}$$

The above term is equal to

$$= \max_{L_p} \left\{ \begin{array}{l} \displaystyle\max_{L_{i+1},\ldots,L_{p-1}} (\sum_{i \leq j \leq p} f_j(L_j) \mid L_i, \ldots, L_p \text{ satisfies } \mathrm{E}(\Gamma)||_{[i..p]}) \\ + \displaystyle\max_{L_{p+1},\ldots,L_{i+k-1}} (\sum_{p \leq j \leq i+k} f_j(L_j) \mid L_p, \ldots, L_{i+k} \text{ satisfies } \mathrm{E}(\Gamma)||_{[p..i+k]}) \\ - f_p(L_p) \end{array} \right\}$$

Now we show that this equals $\displaystyle\max_{L_{i+1},\ldots,L_{i+k-1}} \left\{ \sum_{i \leq j \leq i+k} f_j(L_j) \mid L_i, \ldots, L_{i+k} \text{ satisfies } \mathrm{E}(\Gamma)||_{[i..i+k]}) \right\}$

We first observe that there are no edges from a vertex in the interval $[i+1..p-1]$ to a vertex in the interval $[p+1, i+k]$. This is because we have assumed as input an outer-planar drawing of Γ^{impl}. Therefore, the above two quantities are equal because the maximum for the problem in the region $[i..i+k]$ for a fixed L_i, L_p, L_{i+k} can be obtaining by finding $w(i,p,L_i,L_p)$ and

$w(i, i+k, L_i, L_{i+k})$. We make this statement precise now. Let $L'_{i+1}, \ldots, L'_{p-1}$ be some codon sequences such that $L_i, L'_{i+1}, \ldots, L'_{p-1}, L_p$ satisfies $\mathrm{E}(\Gamma)||_{[i..p]}$, and $L''_{p+1}, \ldots, L''_{i+k-1}$ be a codon sequence such that $L_p, L''_{p+1}, \ldots, L''_{i+k-1}, L_{i+k}$ satisfies $\mathrm{E}(\Gamma)||_{[p..i+k]}$. We have to show that $L_i, L'_{i+1}, \ldots, L'_{p-1}, L_p, L''_{p+1}, \ldots, L''_{i+k-1}, L_{i+k}$ satisfies $\mathrm{E}(\Gamma)||_{[i..i+k]}$. L_i and L_{i+k} are compatible since we have assumed $(i, L_i) \equiv_{E(\Gamma)} (i+k, L_{i+k})$. The compatibility within regions $[i..p]$ and $[p..i+k]$ hold by assumption. The compatibility for vertex pairs (u_r, u_s) with $r \in [i..p-1]$, and the second point is in $s \in [p+1..i+k]$ holds since (i, p) is a maximal edge, and there are no edges in $\mathrm{E}(\Gamma^{\mathrm{impl}})$ between $r \in [i+1..p-1]$ and $s \in [p+1..i+k]$. $\square$