Bioinformatics, 38(16), 2022, 3935–3941
https://doi.org/10.1093/bioinformatics/btac428
Advance Access Publication Date: 28 June 2022
Original Paper

Gene expression

A unifying network modeling approach for codon optimization

Oya Karaşan[1], Alper Şen [1,*], Banu Tiryaki[1] and A. Ercument Cicek [2]

[1]Department of Industrial Engineering, Bilkent University, Ankara 06800, Turkey and [2]Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey

*To whom correspondence should be addressed.

Associate Editor: Lenore Cowen

Received on June 8, 2021; revised on May 1, 2022; editorial decision on June 14, 2022; accepted on June 27, 2022

Abstract

Motivation: Synthesizing genes to be expressed in other organisms is an essential tool in biotechnology. While the many-to-one mapping from codons to amino acids makes the genetic code degenerate, codon usage in a particular organism is not random either. This bias in codon use may have a remarkable effect on the level of gene expression. A number of measures have been developed to quantify a given codon sequence's strength to express a gene in a host organism. Codon optimization aims to find a codon sequence that will optimize one or more of these measures. Efficient computational approaches are needed since the possible number of codon sequences grows exponentially as the number of amino acids increases.

Results: We develop a unifying modeling approach for codon optimization. With our mathematical formulations based on graph/network representations of amino acid sequences, any combination of measures can be optimized in the same framework by finding a path satisfying additional limitations in an acyclic layered network. We tested our approach on bi-objectives commonly used in the literature, namely, Codon Pair Bias versus Codon Adaptation Index and Relative Codon Pair Bias versus Relative Codon Bias. However, our framework is general enough to handle any number of objectives concurrently with certain restrictions or preferences on the use of specific nucleotide sequences. We implemented our models using Python's Gurobi interface and showed the efficacy of our approach even for the largest proteins available. We also provided experimentation showing that highly expressed genes have objective values close to the optimized values in the bi-objective codon design problem.

Availability and implementation: http://alpersen.bilkent.edu.tr/NetworkCodon.zip.

Contact: alpersen@bilkent.edu.tr

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Amino acids in the genetic code are encoded by codons. A codon is a three-letter code in a four-letter (nucleotides Adenine, Thymine, Guanine and Cytosine) alphabet. Because there are 61 possible codons (4^3 minus three stop codons) but only 20 amino acids, the genetic code is said to be degenerate. For example, Leucine can be represented by six different codons: CTT, CTC, CTA, CTG, TTA or TTG. This degeneracy in the genetic code means that the possible number of ways to encode a protein grows exponentially as the number of amino acids in the protein increases. For example, a relatively small size 100-amino acid protein, assuming that each amino acid appears exactly five times, can be expressed by roughly 10^{42} different codon sequences.

While the use of a specific codon or one of its alternatives has no effect on the structure of the protein synthesized, the observed frequencies of such synonymous codons are not equal in nature and vary from one species to the other. For example, Phenylalenine is encoded 46% of the time with TTT and 54% of the time with TTC in *Homo sapiens*; these percentages in *Escherichia coli* are 58% and 42% for TTT and TTC, respectively (Nakamura *et al.*, 2000). This phenomenon is termed as codon-usage bias. Codon usage is often considered to be the most important factor in gene expression (Lithwick and Margalit, 2003). For example, Gustafsson *et al.* (2004) report that using the 'right' codons in the host organism may lead to up to 1000-fold improvements in the expression levels of synthetic genes.

The specific mechanisms by which codon usage affects the protein expression have been an important line of research. The core idea was that translation elongation is rate limited by the tRNA supply and slower rates result from marginal effects of codons that are decoded slowly by relatively rare tRNAs (Brule and Grayhack, 2017). Later,

(i) certain types of wobble decoding (Lareau *et al.*, 2014; Phizicky and Hopper, 2010), (ii) interactions of codons (Buchan *et al.*, 2006) and (iii) the locations of codons in a gene were linked to affect translation rates. Moreover, it was shown that codon choice regulates mRNA levels (Presnyak *et al.*, 2015; Radhakrishnan and Green, 2016) as well as protein folding (Sander *et al.*, 2014) and activity (Xu *et al.*, 2013; Zhou *et al.*, 2013). Reviews by Brar (2016) and Brule and Grayhack (2017) provide more detailed discussion on the mechanisms in codon bias affecting translation rates.

A major consideration in the design of synthetic genes is to adapt the codons to tRNAs that are abundant in the cells of the target organism. In the absence of tRNA abundance information, observed codon frequencies in the target genome are often used as a proxy (Plotkin and Kudla, 2011). There are two main approaches in this line of research. The first one is the *CAI-maximization* approach where the most preferred codons in the target genome are used as much as possible. The second alternative is the *codon sampling* approach. In this alternative, it is argued that consistently using the same (most preferred) codon for one amino acid may often lead to translational errors due to imbalanced use of a subset of the tRNAs (Gustafsson *et al.*, 2004). Therefore, the frequency of the codons used in the gene to be expressed should be similar to the observed frequency of the codons in the target genome.

In addition to codon bias, a similar, but an independent bias, often called codon context bias, is observed when representing adjacent amino acids (i.e. amino acid pairs). An amino acid pair can be represented by as many as 36 different codon pairs. But the observed frequencies of the codon pairs cannot be explained by the observed frequencies of the codons alone (Gutman and Hatfield, 1989). For example, in *Homo sapiens*, the amino acid pair Phenylalenine–Phenylalenine is expected to be represented by the codon pair TTT-TTT with a frequency of 22%, based on the codon frequencies, whereas the actual observed frequency is only 16% and is said to be underrepresented. TTC-TTC pair, on the other hand, is overrepresented with expected and observed frequencies of 28% and 43%, respectively. Coleman *et al.* (2008) have shown that using underrepresented codon pairs leads to reduced levels of gene expression.

In order to quantify how a synthetic gene design complies with the considerations above and to predict the expression level of a gene based on its codon sequence, researchers have developed several measures. One of the first and most widely used measures, Codon Adaption Index (CAI), was developed by Sharp and Li (1987). The index is based on a fitness value (*FV*), which measures how frequent a particular codon expresses an amino acid in comparison to the most frequent codon. For example, the fitness value of codon TTC for Phenylanine, denoted by FV_{TTC}(Phenylanine) is 1, (because TTC is the most frequent codon with a frequency %54), whereas the fitness value of TTT is 0.851 (0.46/0.54). For measuring codon pair bias, Coleman *et al.* (2008) developed a measure based on how much a codon pair is under or overrepresented. Under or overrepresentation of a codon pair is measured using Codon Pair Score (CPS). The Codon Pair Bias is equal to the arithmetic mean of the Codon Pair Scores of each codon pair in the amino acid sequence.

Clarke IV and Clark (2008) introduce a measure called %MinMax to quantify the relative rareness of the codons used for an amino acid sequence in a given window. This is used to identify whether rare (or frequent) codons are clustered in a particular window of the given transcript. Later, a variation of this method is proposed (Wright *et al.*, 2018) and a benchmark that shows the usefulness of this approach compared to other available measures is published (Wright *et al.*, 2020).

In order to measure how the codon choices in a given sequence match the observed frequency of codons in a host organism, Şen *et al.* (2020) developed the Relative Codon Bias (RCB) measure. This distance measure calculates the mean absolute deviations of codon frequencies from observed frequencies for each amino acid and then weighs them by the occurrence of each amino acid in the given sequence. A similar measure, called Relative Codon Pair Bias (RCPB) is defined for codon pairs.

Codon optimization is an approach in synthetic gene design that employs synonymous codon changes in the native gene such that one or more of the measures above (and various other measures) are optimized. The ultimate objective is to improve the level of gene expression in the host organism. The approach uses the codon bias information in the host organism and often also considers various constraints to include or exclude motifs (nucleotide sequences) that may have an effect on expression. Various computational approaches and software tools are suggested for codon optimization, some of which are in use in practice (See Gould *et al.*, 2014 for a review). The majority of these tools consider multiple objectives; see for example Chin *et al.* (2014), which consider four different measures. These tools often use heuristic approaches and the gene designs that they create are not necessarily 'optimal' (Pareto optimal for the multi-objective problems), in the true sense of the word. However, some recent codon optimization approaches provide mathematical guarantees on optimality. Condon and Thachuk (2012) and Arbib *et al.* (2020) study multiple objectives: CAI maximization together with a consideration of excluding or including certain nucleotide sequences; the first study uses a dynamic programming approach, whereas the second one uses a mathematical (mixed integer linear) programming approach. Papamichail *et al.* (2018) study the codon optimization problem when the objectives are CAI and CPB and suggest a polynomial, but an inefficient dynamic programming algorithm to find a design that maximizes CPB subject to CAI not exceeding a particular threshold. The study concludes that the algorithm is not viable for regular sized proteins and suggests a simulated annealing algorithm. Şen *et al.* (2020) show that a mathematical programming approach can be used effectively to find optimal codon designs for two problems: one with objectives CPB and CAI and another with objectives RCB and RCPB. This approach leads to significant improvements in solution time and quality for the first problem over the simulated annealing approach in Papamichail *et al.* (2018). For the first problem (CPB and CAI), Taneda and Asai (2020) develop a very efficient dynamic programming algorithm to generate all Pareto optimal solutions. Though this approach is the fastest available in the literature for the optimization with respect to CPB and CAI measures for small length proteins, it suffers from the high memory requirements of dynamic programming when listing all Pareto optimal (non-dominated) solutions for longer proteins. The approach is also tuned for the mentioned measures and is not flexible enough to handle a third objective or objectives such as RCB and RCPB that depend on the knowledge of the whole codon sequence.

In this study, we model the multi-objective codon optimization problems using network representations of amino acid sequences. We then solve these network problems using mixed integer linear programming. Our mathematical programs benefit greatly from the computational advantages of network models and as such provide the most effective exact solution approaches in handling the measures available in the literature. Moreover, if desired, the proposed approach can handle all the measures jointly within the same unifying framework.

2 Approach and network models

2.1 Measure formulas

Given an amino acid sequence $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)$, the CAI of codon sequence $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ is the geometric mean of the fitness values, given by the following formula.

$$CAI(\gamma, \sigma) = \left(\prod_{i=1}^{n} FV_{\sigma_i}(\gamma_i) \right)^{1/n}.$$

CPS of a codon pair pq for amino acid pair ij is defined as follows.

$$CPS_{pq}(ij) = \ln\left(\frac{\varphi_{pq}\varphi_i\varphi_j}{\varphi_p\varphi_q\varphi_{ij}} \right),$$

where φ denotes the observed frequency of occurrence for an amino acid, codon, amino acid pair or a codon pair. The Codon Pair Bias is formally defined as follows.

$$CPB(\gamma,\sigma)=\frac{\sum_{i=1}^{n-1} CPS_{\sigma_i,\sigma_{i+1}}(\gamma_i,\gamma_{i+1})}{(n-1)}.$$

For a given amino acid sequence in a window that starts at ith amino acid and finishes at kth amino acid, Actual Codon Frequency (ACF_i^k) is the average of the frequencies of codons used for amino acids $\{i,i+1,\ldots,k\}$. Similarly, Maximum Codon Frequency ($MaxCF_i^k$) is the average of the frequencies of codons in the given window when the codons with the maximum frequencies are used and Minimum Codon Frequency ($MinCF_i^k$) is the average of the frequencies of codons in the given window when the codons with the minimum frequencies are used. Finally, Average Codon Frequency ($AvgCF_i^k$) is the average of the average frequencies of codons available for the given window. Then, %MinMax(i, k) is calculated as follows:

$$\begin{cases}\dfrac{ACF_i^k-AvgCF_i^k}{MaxCF_i^k-AvgCF_i^k} & ACF_i^k\geq AvgCF_i^k,\\ \dfrac{AvgCF_i^k-ACF_i^k}{AvgCF_i^k-MinCF_i^k} & ACF_i^k<AvgCF_i^k.\end{cases}$$

The RCB measure is formally defined as follows.

$$\mathrm{RCB}(\gamma,\sigma)=\sum_{i\in\mathcal{A}}\frac{\eta_i(\gamma)}{n}\sum_{p\in\mathcal{C}_i}\frac{1}{|\mathcal{C}_i|}\left|\frac{\vartheta_p(\sigma)}{\eta_i(\gamma)}-\frac{\varphi_p}{\varphi_i}\right|,$$

where $\mathcal{A}=\{\mathcal{A}_1,\ldots,\mathcal{A}_{20}\}$ is the set of amino acids, $\mathcal{C}_i=\{C_{i_1},\ldots,C_{i_{|\mathcal{C}_i|}}\}$ is the set of codons that can be used to represent amino acid $i\in\mathcal{A}$, $\eta_i(\gamma)$ is the number of times amino acid i appears in the sequence γ and $\vartheta_p(\sigma)$ is the number of times codon p appears in the codon sequence σ.

The RCPB is defined as follows.

$$\mathrm{RCPB}(\gamma,\sigma)=\sum_{i,j\in\mathcal{A}}\frac{\eta_{ij}(\gamma)}{n-1}\sum_{p\in\mathcal{C}_i,q\in\mathcal{C}_j}\frac{1}{|\mathcal{C}_i||\mathcal{C}_j|}\left|\frac{\vartheta_{pq}(\sigma)}{\eta_{ij}(\gamma)}-\frac{\varphi_{pq}}{\varphi_{ij}}\right|,$$

where $\eta_{ij}(\gamma)$ and $\vartheta_{pq}(\sigma)$ are defined similarly for amino acid pairs and codon pairs.

In order to illustrate the impact of codon design on these measures, we provide an example of a small protein (peptide) whose amino acid sequence is CCHC where C stands for Cysteine and H stands for Histidine (see Fig. 1). Cysteine can be expressed by codons TGC and TGT, Histidine can be expressed by codons CAC and CAT. This means that CCHC can be expressed by 16 different codon sequences. Frequency (φ_p/φ_i) and fitness values for codons and frequency ($\varphi_{pq}/\varphi_{ij}$) and CPS information for codon pairs in humans are provided in Table 1 [sources: Nakamura *et al.* (2000) for codons and Coleman *et al.* (2008) for codon pairs].

Using codon sequence TGC-TGC-CAT-TGC (represented by the solid line in Fig. 1) leads to CAI, CPB, RCB and RCPB scores of 0.720, -0.038, 0.487 and 0.357, respectively. Using the codon sequence TGC-TGT-CAC-TGT (represented by the dotted line) instead leads to CAI, CPB, RCB and RCPB scores of 0.704, -0.022, 0.263 and 0.369. The second sequence is worse in the use of more frequent codons and in matching the observed codon pair frequencies, but is better in terms of the use of more frequent codon pairs and in matching the observed codon frequencies. The example clearly shows that there is a trade-off between various measures that may have effects on protein expression.

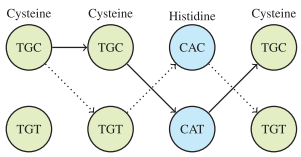

Fig. 1. A network representation for peptide CCHC. There are a total of 16 possible codon sequence choices. Using codon sequence represented by the solid lines leads to CAI, CPB, RCB and RCPB scores of 0.720, –0.038, 0.487 and 0.357, respectively. Using the sequence of the dotted lines instead leads to CAI, CPB, RCB and RCPB scores of 0.704, –0.022, 0.263 and 0.369

2.2 Preliminaries on optimization

Optimization or mathematical programming problems consist of maximizing or minimizing real valued objective functions of decision variables from among an allowed set of domain values. The codon optimization models we propose in this article are going to be mixed integer linear programming (MILP) models of the form min $\mathbf{c}^T\mathbf{x}$ s.t. $\mathbf{Ax}\leq\mathbf{b}$ where $\mathbf{c}$ is a vector in $\mathbb{R}^{m+n}$, $\mathbf{b}$ is a vector in $\mathbb{R}^p$, $\mathbf{A}$ is a $p\times(m+n)$ matrix and decision variables $\mathbf{x}\in\mathbb{Z}^m\times\mathbb{R}^n$. While MILP is an NP-Hard problem in general, scientifically developed commercial solvers are in use for many successful real-life applications. Most of these solvers utilize the underlying structural properties of the constraint matrix, $\mathbf{A}$. One of the most successful applications of MILP is when there is an inherent network structure in the models (Ahuja *et al.*, 1993; Olson, 2003). To this end, we revisit the existing MILP approaches available in the literature (Arbib *et al.*, 2020; Şen *et al.*, 2020) and propose novel models that incorporate a network structure. In order to benefit from the computational tractability (in practice) of MILP models with underlying network structures, we unify existing codon optimization measures in the literature under the network optimization framework and cast the codon optimization problem as a path finding problem in a layered acyclic network. Though, the fundamental algorithm behind the state of the art MILP solvers, namely, branch and bound, has exponential space and time complexity, we are able to show that codon optimization problems utilizing common measures of the literature can be solved for realistic dimensions in reasonable times. For theoretical completeness, we also show that when certain pairs of measures are utilized within a bi-objective framework, dynamic programming algorithms can be designed to solve the resulting codon optimization problem in polynomial time complexity, albeit with very large degrees of polynomial. Note that although the dynamic programming algorithms are designed to handle only two measures simultaneously, the proposed network models can be adapted to handle any combination of the measures available in the literature.

The following is necessary to provide the required formalism. Let $\gamma=\{\gamma_1,\ldots,\gamma_n\}$ be the given sequence of amino acids and assume $\pi:\{1,\ldots,n\}\to\{1,\ldots,20\}$ provides the many to one mapping such that the ith amino acid in the sequence is $\mathcal{A}_{\pi_i}$. We shall use a layered directed network to represent our codon optimization problems. Each layer in this network will correspond to an amino acid in our protein and the nodes in this layer will correspond to codons that can be used in the expression of this amino acid. Let $\mathcal{C}_i=\{C_{i_1},\ldots,C_{i_{|\mathcal{C}_i|}}\}$ be the set of codons that can be used in expressing amino acid $i\in\mathcal{A}$. We know that $|\mathcal{C}_i|\leq 6$ for each $i\in\{1,\ldots,20\}$ and the set $\mathcal{C}=\cup_{i=1}^{20}\mathcal{C}_i$ has 61 codons. Each codon $p\in\mathcal{C}$ is identified with a unique amino acid and we let $\mathcal{A}(p)$ denote this amino acid. Our layered network will have the node set $V=\{s,t\}\cup\cup_{i=1}^{n}V_i$ where $V_i=\left\{\sum_{j=1}^{i-1}|\mathcal{C}_{\pi_j}|+1,\ldots,\sum_{j=1}^{i-1}|\mathcal{C}_{\pi_j}|+|\mathcal{C}_{\pi_i}|\right\}$ and s and t are the source and destination nodes, respectively. For notational convenience let $V_0=$

Table 1. Frequency information related to peptide CCHC

AA	Codon	Freq	Fitness V.	AA pair	Codon pair	Freq	CPS
C	TGC	0.544	1.000	C-C	TGC-TGC	0.501	0.531
	TGT	0.456	0.839		TGC-TGT	0.302	0.195
H	CAC	0.581	1.000		TGT-TGT	0.102	–0.721
	CAT	0.418	0.720		TGU-TGC	0.095	–0.956
				C-H	TGC-CAC	0.417	0.281
					TGC-CAT	0.244	0.065
					TGT-CAT	0.162	–0.174
					TGT-CAC	0.178	–0.398
				H-C	CAC-TGC	0.465	0.391
					CAC-TGT	0.304	0.136
					CAT-TGT	0.119	–0.479
					CAT-TGC	0.112	–0.710

$\{s\}$ and $V_{n+1} = \{t\}$. There will be an arc between any two nodes in consecutive layers in this network, i.e. the arc set is $E = \{(k,l) : k \in V_i, l \in V_{i+1}, i = 0,\ldots,n\}$. We represent our layered network as $G = (V,E)$. For each node $k \in V$ we let $L(k)$ denote the layer or equivalently the sequence of the corresponding amino acid for this node. In particular, if $k \in V_i$ then $L(k) = i$. Each node $k \in V \setminus \{s,t\}$ corresponds to a particular amino acid and a particular codon, which we define by $A(k)$ and $C(k)$, respectively. In particular, $A(k) = \mathcal{A}_{\pi_{L(k)}}$ and $C(k) = \mathcal{C}_{A(k)_q}$ where $q = k - \sum_{i=1}^{L(k)-1} |V_i|$. An example network is provided in Supplementary S1.

2.3 Codon pair bias and codon adaptation index

We first present our network framework for the most commonly adopted measures in the literature, namely, CAI and CPB. To find the codon sequences with CAI and CPB values that are not jointly dominated, the epsilon-constraint method widely used for solving bi-objective problems in the literature (see Laumanns *et al.*, 2006) is utilized. The method is based on solving single-objective models iteratively, limiting the second objective value by a constraint. To find the efficient frontier corresponding to non-dominated CAI and CPB values, among alternative solutions with the same CAI (CPB) value, the one that gives the highest CPB (CAI) value should be found. At each iteration of the epsilon-constraint method, two models are solved; one maximizing CPB among alternative solutions achieving a particular CAI value (say minCAI) and another doing the opposite. The problem of finding a codon sequence maximizing CPB while imposing a restriction on CAI is called CPB versus CAI.

For each $(k,l) \in E$, we use binary decision variable $x_{kl} = 1$, if amino acid $A(k)$ is represented by codon $C(k)$ and amino acid $A(l)$ is represented by codon $C(l)$; 0, otherwise. The cost of this arc is the Codon Pair Score of the corresponding codon pair, i.e. $CPS_{C(k)C(l)}$. Each node $k \in V \setminus \{s,t\}$ also has a coefficient representing the natural logarithm of the Fitness Value of the corresponding codon, i.e. $\ln(FV_{C(k)})$. Our model for solving CPB versus CAI is the following.

$$\max \sum_{(k,l)\in E: k\neq s, l\neq t} \frac{1}{n-1} CPS_{C(k)C(l)} x_{kl} \tag{1}$$

$$\text{s.t.} \sum_{\{l:(k,l)\in E\}} x_{kl} - \sum_{\{l:(l,k)\in E\}} x_{lk} = \begin{cases} 1, & k = s \\ -1, & k = t \\ 0, & \text{otherwise} \end{cases} \qquad k \in V, \tag{2}$$

$$\sum_{(k,l)\in E: k\neq s} \ln(FV_{C(k)}) x_{kl} \geq n \ln(\text{minCAI}), \tag{3}$$

$$x_{kl} \in \{0,1\} \quad (k,l) \in E. \tag{4}$$

Objective function (1) ensures that the CPB of the codon sequence is maximized. Constraints (2) construct a path from s to t in the network representation of the underlying amino sequence. The nodes chosen on this path will simply correspond to the optimized codons in the given sequence. Constraint (3) uses the fact that $\ln(CAI(\gamma,\sigma)) = \frac{1}{n}\sum_{i=1}^{n} \ln(FV_{\sigma_i}(\gamma_i))$ and ensures that the CAI of the selected codon sequence is greater than or equal to a predetermined minimum Codon Adaptation Index value. Finally, constraints (4) ensure that all the decision variables are binary.

It is possible to come up with a dynamic programming algorithm of polynomial complexity to solve the above bi-objective optimization problem. Though similar results have been established in Papamichail *et al.* (2018) and Donoghue *et al.* (2017), we choose to provide it here along with its proof in Supplementary S2 as it sets the framework for our other complexity results to follow.

Theorem 1.*Problem CPB versus CAI can be solved in polynomial time.*

2.4 Relative codon pair bias and relative codon bias

We now turn our attention to using RCPB and RCB objectives jointly as the gene expression level measures. We apply epsilon-constraint method to minimize both objectives jointly. To this end, we solve a sequence of Mixed Integer Programming (MIP) models minimizing RCPB (RCB) by imposing a threshold value on RCB (RCPB). By ranging the threshold values, we attain the Pareto optimal set of solutions, i.e. the efficient frontier. The details on the mathematical models along with complexity results are provided in Supplementary S3.

The methodology used for optimizing CPB and CAI measures can be easily adapted to the measures based on sliding windows such as %MinMax introduced by Clarke IV and Clark (2008). For details, the interested reader is referred to Supplementary S4. In addition to optimizing a measure, it is often important to avoid or include certain nucleotide sequences to promote gene expression. For example, certain nucleotide sequences may include restriction enzyme recognition sites of the host organisms and should be avoided (Skiena, 2001), whereas certain nucleotide sequences are immunostimulatory and should be included as much as possible in the synthesized gene (Condon and Thachuk, 2012). The nucleotide sequences that need to be avoided are called forbidden motifs, and the nucleotide sequences that need to be included are called desired motifs. Details on how to incorporate such forbidden and desired motifs in our models can be found in Supplementary S5.

2.5 Mathematical programming versus dynamic programming

Though all our problems can be solved in polynomial time, the exponents in the polynomial functions (though overestimated for simplicity of the arguments), are huge. Thus, dynamic programming algorithms will not be viable options due to memory requirements for realistic dimensions. We would like to stress that our models based on network flows tackle this challenge very effectively and provide the flexibility of using any combination of the discussed measures even under forbidden and desired motif requirements.

3 Implementation and results

3.1 Computational efficacy of the network models

We solved CPB versus CAI and RCPB versus RCB models using Gurobi Optimizer 9.1.1 (Gurobi Optimization, 2022) with Python 2.7.13 on an Intel Core i7-3820 3.6 GHz CPU. In our experimental design, we used Codon Pair Scores, Fitness Values, Relative Codon Pair Bias values and Relative Codon Bias values of *Homo sapiens*. To show that our methodology can scale well to all dimensions in the available datasets, we analyzed a random sample of 100 proteins from the UniProt Database (The UniProt Consortium, 2019) for both problems. For the CPB versus CAI problem, we additionally analyzed the 10 biggest proteins to demonstrate that classical objectives can be easily handled for even the largest proteins available. In order to compare our methodology with the literature in terms of solution times, we also solved both problems using the mathematical programming approach by Şen *et al.* (2020) with the same proteins.

For the CPB versus CAI problem, for each protein, we ran both the existing and proposed models for minCAI values ranging between 0.55 and 1 with 0.05 increments. Figure 2 depicts an efficient frontier for Codon Pair Bias and Codon Adaptation Index. The analysis is done with the largest protein, Titin, which consists of 34 350 amino acids. In particular, the point (0.66, 0.38) indicates that 0.66 is the greatest CAI value that satisfies a minimum CPB value of 0.38 and 0.38 is the greatest CPB value that satisfies a minimum CAI value of 0.66. To find each point in Figure 2, model (1)–(4) is ran two times by switching the objectives. In Table 2, we present the results for proteins of different sizes. The first column reports the interval for the sizes of the proteins in the number of amino acids and the second column reports the total number of instances (each protein-minCAI pair is an instance). For each protein, we solve the problem for nine different minCAI values in set $\{0.55, 0.60, \ldots, 0.95\}$. For each interval, we present the average solution times, the maximum solution times and the standard deviations in seconds based on the proposed network formulation [model (1)–(4)] and the existing formulation of Şen *et al.* (2020). The average time gain column represents how much faster our proposed formulation is compared to the existing

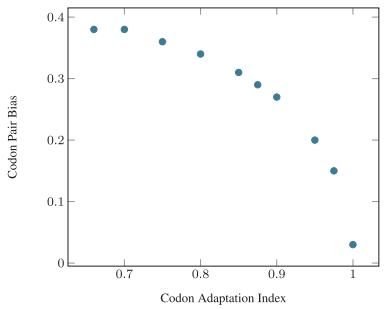

Fig. 2. An efficient frontier representing the relation between CAI and CPB for a protein containing 34 350 amino acids. The analysis is done in the range of maximum possible CPB of the protein and the maximum CAI value. Each point in the figure is obtained by solving the model (1)–(4) two times by switching the objectives between CAI and CPB

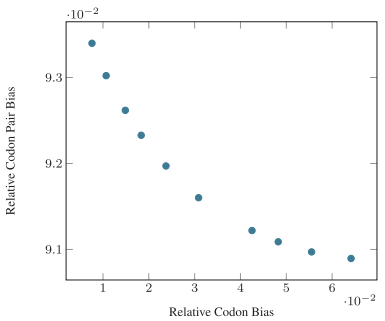

Fig. 3. An efficient frontier representing the relation between RCB and RCPB for a protein containing 824 amino acids. Each point in the figure is obtained by solving the model [Supplementary S3: (6)–(13)] two times by switching the objectives between RCB and RCPB

formulation, attained through averaging the formula TimeGain $= \frac{t_{\text{exist}} - t_{\text{proposed}}}{t_{\text{exist}}} \times 100$ for each instance within the group. The results show that our proposed formulation is very strong and leads to drastic time gains over the approach used in Şen *et al.* (2020). Most problems can be solved within a second; the largest protein requires a maximum of only about seven minutes of solution time. A comprehensive computational analysis is provided in Supplementary S6.

For the RCPB versus RCB problem, we solved both the proposed [Supplementary S3: (6)–(13)] and the existing [model of Şen *et al.* (2020)] models for minRCB values ranging between 0.01 and 0.1 with 0.01 increments. Figure 3 depicts an efficient frontier for RCPB and RCB. The analysis is done with a protein with 824 amino acids. We present the solution times in seconds for the RCPB versus RCB problem in Table 3. The numbers in parentheses in the second column show the number of instances that turned out to be feasible within this group. With the proposed formulation, one instance in 800–899 and 900–999 length intervals and 8 instances in 1000–1365 length interval could not be solved in 3600 seconds time limit. With the existing formulation, there are 2 instances in 600–699, 10 instances in 800–899, 6 instances in 900–999 and 26 instances in 1000–1365 length intervals, respectively that could not be solved within the given time limit. These instances are not included in the calculations. As the last column clearly shows, the network modeling approach drastically reduced the solution times and this improvement is more pronounced for larger proteins. We note that this problem is more challenging than the CPB versus CAI problem. We depict the same data highlighting the effect of maxRCP constraint and the length of the sequence on the solution times in Supplementary S7.

3.2 Natural designs versus proposed measures

In order to further investigate our approach and evaluate the performance of the proposed measures in terms of their effect on gene expression, we analyzed codon sequences of genes of five different species: *E.coli*, *Arabidopsis thaliana*, *Mus musculus*, *Caenorhabditis elegans* and *Saccharomyces cerevisiae*. For all codon sequences available in the dataset provided by Sterken *et al.* (2020), we measured CAI, CPB, RCB and RCPB values using codon and codon pair frequencies from HIVE database (Alexaki *et al.*, 2019). We use transcript abundance (TA) values presented by Sterken *et al.* (2020) in order to measure the level of gene expression. For each organism, we find the 5% of the genes having the highest TA values ('highly expressed genes') and the 5% of the genes having the lowest TA values ('lowly expressed genes'). In Table 4, for each measure and for each organism, we report the average values of the measure for the lowly expressed genes (first row) and the highly expressed genes (second row) and the *P*-value for the *t*-test for the difference of mean values (third row). *n* in this table corresponds to the number of genes that make up the 5% of the genes for a given organism. The results show that CAI values are significantly higher ($P = 0.01$) for highly expressed genes for *E.coli*, *S.cerevisiae* and *M.musculus*. CPB values are significantly higher for *C.elegans*, *S.cerevisiae* and *A.thaliana*, but significantly lower for *E.coli*. RCB values are significantly higher for all organisms. RCPB does not seem to be significantly different for the highly and lowly expressed genes. We also ran permutation tests to see the impact of different measures on TA (Table 5). For this purpose, for each species, we create 100 samples, each consisting of a random 5% of the genes. For each sample, we find the average value of each measure. We then compare these values with the average values we get from the highly expressed genes. The average value of CAI of the highly expressed genes is larger than the average value of CAI for all 100 random samples for *E.coli*, *C.elegans* and *S.cerevisiae*. For *M.musculus* and *A.thaliana*, this number was 34 and 98, respectively. The average value of CPB for the highly expressed genes is larger than the average value of CPB for all 100 random samples for *C.elegans*, *S.cerevisiae* and *A.thaliana*. Interestingly, the average CPB for the highly expressed genes is *smaller* than all 100 random samples for *E.coli* and *M.musculus*. For all species, the average values of RCB and RCPB of the highly expressed genes are larger than the corresponding averages in all 100 samples.

Based on these analyses, one can see that CAI and CPB measures both play a role in expression for *C.elegans*, *S.cerevisiae* and *A.thaliana* and the dual objective approach presented in Section 2.3 may be needed. For these species, we analyze 5% of the highest expressed genes and 5% of the lowest expressed genes and calculate the efficient frontier for each gene using the model in Section 2.3. We then calculate the distance of each gene (as defined in the original codon sequence) from its efficient frontier of CAI and CPB. In doing this, we normalize the CAI and CPB values by calculating their *z*-scores. For example, for a given organism, the *z*-score for a particular gene's CAI value is its CAI value minus the average CAI value of all genes for that organism divided by the standard deviation of the CAI values. We then find the point in the efficient frontier that has the smallest Euclidean distance to the corresponding gene's natural transcript. For all three species, the mean value of Euclidean distances to the efficient frontier for the highly expressed genes are significantly smaller than those for the lowly expressed genes (the *P*-values are 0.0005, 10^{-92} and 0.0909 for *C.elegans*, *S.cerevisiae* and *A.thaliana*, respectively). For *S.cerevisiae*, the Euclidean distances for the highly and lowly expressed genes are plotted in Supplementary S8. We also checked to see whether one of the measures (CAI or CPB) is the dominant measure in the highly expressed genes, perhaps leading one to consider only a single measure (objective) when optimizing codon sequences. This is not the case. For example, for *A.thaliana*, normalized CAI values for the 5% of the highly expressed genes range between –2.689 and 2.851 and normalized CPB values range between –3.601 and 5.092. For highly expressed genes with CAI in range $(-2.689, -0.842)$ (first third of the full range), the average normalized CPB value is 1.042; for genes in range $(-0.842, 1.004)$ (second third), the average normalized CPB is 0.256; for genes in range $(1.004, 2.851)$ (last third), the average normalized CPB is 0.194. Clearly, CPB is decreasing as CAI is increasing in highly expressed genes. In fact, there is a negative correlation between CAI and CPB values for the highly expressed genes (Pearson correlation coefficient is -0.191 with *P*-value 0.00019). These analyses show that CAI and CPB measures

Table 2. CPB versus CAI solution time statistics for proposed and existing formulations for the random sample of 100 proteins

Length	#	Proposed			Existing			Avg.
		Avg. Time	Max. Time	Std. Dev.	Avg. Time	Max. Time	Std. Dev.	Time Gain
0–99	63	0.02	0.06	0.01	0.13	0.31	0.07	85.42
100–199	171	0.04	0.10	0.02	0.29	0.56	0.09	86.10
200–299	198	0.07	0.12	0.02	0.60	1.12	0.17	87.91
300–399	162	0.11	0.21	0.03	1.11	1.73	0.29	89.91
400–499	90	0.13	0.21	0.04	1.42	2.25	0.36	90.43
500–599	54	0.17	0.24	0.04	2.11	2.95	0.51	91.87
600–699	45	0.21	0.35	0.05	2.97	4.55	0.81	92.88
700–799	9	0.20	0.30	0.05	3.37	4.21	0.62	93.91
800–899	45	0.29	0.48	0.07	4.67	6.16	0.72	93.78
900–999	18	0.31	0.55	0.09	5.47	7.15	0.75	94.34
1000–1365	45	0.38	0.71	0.11	6.68	11.23	1.72	94.12
6669	9	2.41	3.52	0.82	87.31	131.96	26.0	97.21
6885	9	7.16	11.01	3.02	111.62	157.62	32.41	93.00
6907	9	8.27	11.60	2.42	108.53	157.62	32.02	92.15
7388	9	3.71	4.99	1.01	136.18	193.13	40.53	97.12
7570	9	11.17	16.01	3.96	130.61	192.46	37.60	91.30
7968	9	12.17	20.12	4.52	189.46	278.64	54.76	93.35
8384	9	50.99	78.03	13.41	399.34	612.46	161.54	84.26
8797	9	5.99	8.29	1.66	175.10	241.92	55.0	96.33
14507	9	35.34	75.47	17.30	723.53	1029.41	257.60	94.77
34350	9	159.74	426.92	114.76	1755.59	2499.73	582.28	90.87

Notes: For each protein both models are solved for 9 different CAI values between 0.55 and 0.95 with 0.05 increments. The table represents the average and maximum solution times and the standard deviations in seconds for different length intervals and for the largest 10 proteins. The average time gain column represents how much faster our proposed formulation is compared to the existing formulation.

Table 3. RCPB versus RCB solution time statistics for proposed and existing formulations for the random sample of 100 proteins

Length	#	Proposed			Existing			Avg.
		Avg. Time	Max. Time	Std. Dev.	Avg. Time	Max. Time	Std. Dev.	Time Gain
0–99	70 (22)	0.35	0.84	0.20	1.43	3.95	0.84	74.78
100–199	190 (131)	0.53	3.55	0.50	3.94	29.22	4.0	84.94
200–299	220 (189)	1.11	5.40	0.93	10.07	52.72	8.53	87.94
300–399	180 (164)	4.90	229.58	18.05	39.31	716.49	62.02	89.99
400–499	100 (95)	6.67	76.77	10.83	71.66	904.03	108.26	90.54
500–599	60 (60)	34.12	855.18	113.76	228.41	3302.33	438.39	90.33
600–699	50 (50)	51.94	501.54	92.63	405.20[a]	3545.63	622.59	89.18
700–799	10 (10)	74.38	423.20	123.79	634.62	2202.07	673.91	89.60
800–899	50 (50)	87.96[a]	350.84	74.26	1200.64[a]	3597.85	749.17	92.74
900–999	20 (20)	87.10[a]	243.98	67.27	1372.42[a]	2991.93	657.90	94.03
1000–1365	50 (50)	111.40[a]	344.31	96.56	1767.71[a]	3593.93	1201.45	94.29

Notes: For each protein, both models are solved for 10 different RCB values between 0.01 and 0.1 with 0.01 increments. The table represents the average and maximum solution times together with the standard deviations in seconds for different length intervals. The numbers in parentheses in the second column represents the number of instances that turn out to be feasible. Only the feasible instances are included in the calculations.

[a]In length intervals 800–899 and 900–999, one instance could not be solved within 3600 seconds and in length interval 1000–1365, 8 instances could not be solved within the given time limit with the new formulation. With the existing formulation, there are 2 instances in 600–699 length interval, 10 instances in 800–899 length interval, 6 instances in 900–999 length interval and 26 in 1000–1365 length interval that could not be solved within 3600 seconds. Only the instances that are solved within the 3600 seconds time limit with both formulations are used for average time, maximum time, standard deviation and average time gain calculations.

are often in conflict and one may have to consider designs on and around the efficient frontier when designing codon sequences. This justifies the dual objective approach.

4 Conclusions

We present a flexible and effective methodology for codon optimization. Our novel approach can handle any combination of objectives measuring gene expression levels in a unifying framework and is more effective than the existing methods. The basis is a network representation of the given protein. Any measure, including but not restricted to the ones available in the literature, which depends on a codon sequence can be handled with our methodology. Moreover, the bi-objective limitation of the existing dynamic programming methods in the literature is not present with our approach that has the ability to tackle any number of objectives. The software we developed allows one to choose any objective and use any subset of other objective limitations as constraints. For future work, we plan to incorporate other specific measures and constraints that are

Table 4. CAI, CPB, RCB and RCPB values for highly and lowly expressed genes

	E.coli	*C.elegans*	*S.cerevisiae*	*M.musculus*	*A.thaliana*
n	121	376	281	896	419
CAI	0.6868	0.7617	0.7247	0.7773	0.7638
	0.7679	0.7657	0.8112	0.7823	0.7648
	0.0000	0.1269	0.0000	0.0057	0.6957
CPB	0.0468	0.0317	0.0092	0.0642	0.0342
	0.0195	0.0511	0.0421	0.0602	0.0501
	0.0000	0.0000	0.0000	0.0166	0.0000
RCB	0.1235	0.1016	0.1160	0.1084	0.1034
	0.1688	0.1644	0.2207	0.1216	0.1314
	0.0000	0.0000	0.0000	0.0000	0.0000
RCPB	0.1614	0.1634	0.1700	0.1553	0.1699
	0.1639	0.1680	0.1681	0.1658	0.1683
	0.3456	0.0118	0.2957	0.0000	0.2240

Note: The analysis is done for species *E.coli*, *A.thaliana*, *M.musculus*, *C.elegans* and *S.cerevisiae*. High and low gene expression levels are based on TA values of the genes. The average CAI, CPB, RCPB and RCB values are calculated for 5% of the highly expressed genes and 5% of the lowly expressed genes for each species. Each cell is about a measure and an organism. The first and the second numbers are the average value of the measure for lowly expressed and highly expressed genes, respectively. The third number is the *P*-value of the *t*-test.

Table 5. Permutation analyses for highly and lowly expressed genes

	E.coli	*C.elegans*	*S.cerevisiae*	*M.musculus*	*A.thaliana*
CAI	0.01	0.01	0.01	0.67	0.03
CPB	1.00	0.01	0.01	1.00	0.01
RCB	0.01	0.01	0.01	0.01	0.01
RCPB	0.01	0.01	0.01	0.01	0.01

shown to affect gene expression using to our model. We also plan to study the importance of various measures on other organisms such as mammals where gene expression is more critical for survival.

Financial Support: none declared.

Conflict of Interest: none declared.

References

Ahuja,R.K. *et al.* (1993) *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, New Jersey.

Alexaki,A. *et al.* (2019) Codon and codon-pair usage tables (CoCoPUTs): facilitating genetic variation analyses and recombinant gene design. *J. Mol. Biol.*, **431**, 2434–2441.

Arbib,C. *et al.* (2020) Codon optimization by 0-1 linear programming. *Comput. Oper. Res.*, **119**, 104932.

Brar,G.A. (2016) Beyond the triplet code: context cues transform translation. *Cell*, **167**, 1681–1692.

Brule,C.E. and Grayhack,E.J. (2017) Synonymous codons: choose wisely for expression. *Trends Genet.*, **33**, 283–297.

Buchan,J.R. *et al.* (2006) tRNA properties help shape codon pair preferences in open reading frames. *Nucleic Acids Res.*, **34**, 1015–1027.

Chin,J.X. *et al.* (2014) Codon Optimization OnLine (COOL): a web-based multi-objective optimization platform for synthetic gene design. *Bioinformatics*, **30**, 2210–2212.

Clarke,T.F. IV and Clark,P.L. (2008) Rare codons cluster. *PLoS One*, **3**, e3412.

Coleman,J.R. *et al.* (2008) Virus attenuation by genome-scale changes in codon pair bias. *Science*, **320**, 1784–1787.

Condon,A. and Thachuk,C. (2012) Efficient codon optimization with motif engineering. *J. Discrete Algorithms*, **16**, 104–112.

Donoghue,N. *et al.* (2017) Optimal codon pair bias design (extended abstract). In: *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Kansas City*, IEEE, pp. 529–532.

Gould,N. *et al.* (2014) Computational tools and algorithms for designing customized synthetic genes. *Front. Bioeng. Biotechnol.*, **2**, 41.

Gurobi Optimization,LLC. (2022) *Gurobi Optimizer Reference Manual* https://www.gurobi.com.

Gustafsson,C. *et al.* (2004) Codon bias and heterologous protein expression. *Trends Biotechnol.*, **22**, 346–353.

Gutman,G.A. and Hatfield,G.W. (1989) Nonrandom utilization of codon pairs in *Escherichia coli*. *Proc. Natl. Acad. Sci. USA*, **86**, 3699–3703.

Lareau,L.F. *et al.* (2014) Distinct stages of the translation elongation cycle revealed by sequencing ribosome-protected mRNA fragments. *Elife*, **3**, e01257.

Laumanns,M. *et al.* (2006) An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *Eur. J. Oper. Res.*, **169**, 932–942.

Lithwick,G. and Margalit,H. (2003) Hierarchy of sequence-dependent features associated with prokaryotic translation. *Genome Res.*, **13**, 2665–2673.

Nakamura,Y. *et al.* (2000) Codon usage tabulated from international DNA sequence databases: status for the year 2000. *Nucleic Acids Res.*, **28**, 292–292.

Olson,D.L. (2003) Optimization models. In: Bidgoli,H. (ed.) *Encyclopedia of Information Systems*. Elsevier, New York, pp. 403–411.

Papamichail,D. *et al.* (2018) Codon context optimization in synthetic gene design. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **15**, 452–459.

Phizicky,E.M. and Hopper,A.K. (2010) tRNA biology charges to the front. *Genes Dev.*, **24**, 1832–1860.

Plotkin,J.B. and Kudla,G. (2011) Synonymous but not the same: the causes and consequences of codon bias. *Nat. Rev. Genet.*, **12**, 32–42.

Presnyak,V. *et al.* (2015) Codon optimality is a major determinant of mRNA stability. *Cell*, **160**, 1111–1124.

Radhakrishnan,A. and Green,R. (2016) Connections underlying translation and mRNA stability. *J. Mol. Biol.*, **428**, 3558–3564.

Sander,I.M. *et al.* (2014) Expanding Anfinsen's principle: contributions of synonymous codon selection to rational protein design. *J. Am. Chem. Soc.*, **136**, 858–861.

Şen,A. *et al.* (2020) Codon optimization: a mathematical programming approach. *Bioinformatics*, **36**, 4012–4020.

Sharp,P.M. and Li,W.-H. (1987) The codon adaptation index-a measure of directional synonymous codon usage bias, and its potential applications. *Nucleic Acids Res.*, **15**, 1281–1295.

Skiena,S.S. (2001) Designing better phages. *Bioinformatics*, **17**, S253–S261.

Sterken,M.G. *et al.* (2020) Conserved codon adaptation in highly expressed genes is associated with higher regularity in mRNA secondary structures. *bioRxiv*.

Taneda,A. and Asai,K. (2020) Cosmo: a dynamic programming algorithm for multicriteria codon optimization. *Comput. Struct. Biotechnol. J.*, **18**, 1811–1818.

The UniProt Consortium. (2019) UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res.*, **47**, D506–D515.

Wright,G. *et al.* (2018) HarMinMax: harmonizing codon usage to replicate local host translation. In: *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, *Washington, DC*, ACM, pp. 528–528.

Wright,G. *et al.* (2020) Analysis of computational codon usage models and their association with translationally slow codons. *PLoS One*, **15**, e0232003.

Xu,Y. *et al.* (2013) Non-optimal codon usage is a mechanism to achieve circadian clock conditionality. *Nature*, **495**, 116–120.

Zhou,M. *et al.* (2013) Non-optimal codon usage affects expression, structure and function of clock protein FRQ. *Nature*, **495**, 111–115.