

Comparing Metaheuristics for Semantics-Preserving Readability Optimization

Jorge Martinez-Gil

Software Competence Center Hagenberg GmbH, Softwarepark 32a, Hagenberg, 4232, Austria

Abstract

Automatic readability optimization rewrites a document to a lower reading grade by substituting hard words with simpler synonyms. Prior work cast this as a single-objective search solved with a genetic algorithm, but making a text easier, preserving its meaning, and changing as few words as possible are three competing goals that a single scalar objective cannot express. We reframe the task as a multi-objective optimization problem, minimizing a readability grade, an edit rate, and a semantic distance simultaneously, and turn it into a reproducible benchmark for comparing metaheuristics. On a ten-document corpus, with an identical lexical resource and an equal budget of evaluations, we compare eight optimizers: random search, a genetic algorithm, three single-objective metaheuristics (Jaya, Cuckoo Search, TLBO) applied to a scalarized objective, and three dedicated multi-objective evolutionary algorithms (NSGA-II, GDE3, SMPSO). The multi-objective optimizers significantly outperform the rest on hypervolume (Friedman test, $p = 1.6 \times 10^{-11}$; GDE3 first, then SMPSO and NSGA-II) and, crucially, recover the whole readability/meaning/edit trade-off curve, including the low-edit region a practitioner would deploy, whereas scalarizing methods concentrate on the aggressive-simplification extreme and over-edit the text by a factor of three to twenty. An ablation shows that adding an explicit semantic objective removes almost all unnecessary editing and meaning drift at negligible cost in readability. The benchmark is training-free, deterministic, transparent and controllable, providing an auditable baseline that complements large language model simplification.

Keywords: readability optimization, text simplification, multi-objective optimization, metaheuristics, evolutionary computation, benchmark

1. Introduction

A large share of the documents that citizens must read every day, health leaflets, legal contracts, government forms, financial disclosures, educational material and scientific abstracts, are written well above the reading level of their intended audience. Automatic text simplification aims to rewrite such documents so that they are easier to read while preserving their meaning, and has become a central task in natural language processing for accessibility and social good [1].

Two broad families of methods dominate the field. The first is data-driven rewriting, culminating in sequence-to-sequence and, more recently, large language model (LLM) approaches that generate a simplified paraphrase of the input. These methods are fluent but opaque: the trade-off they strike

Email address: `jorge.martinez-gil@scch.at` (Jorge Martinez-Gil)

between how much easier the text becomes, how much of its meaning is retained, and how much of it is actually changed is *implicit*, uncontrollable, and difficult to reproduce across model versions and prompts. The second family treats simplification as an explicit optimization problem over a discrete space of edits. [2] recently framed lexical readability optimization as a search problem and solved it with a single-objective genetic algorithm (GA), replacing hard words by simpler synonyms so as to minimize a readability formula while attempting to preserve the content of the text.

The single-objective view, however, hides a fundamentally multi-criteria decision. Making a text easier to read (lower grade level), preserving its meaning, and changing as few words as possible are three *competing* goals: aggressively lowering the grade level tends to require many substitutions and to drift away from the original meaning, whereas a conservative rewrite preserves meaning perfectly but barely improves readability. Collapsing these objectives into a single weighted score forces the practitioner to commit to a weighting *before* seeing the alternatives, and, as we show systematically biases the search toward one end of the trade-off.

In this paper we reframe semantics-preserving readability optimization as a *multi-objective* problem and turn it into a reproducible benchmark for comparing metaheuristics. Concretely, our contributions are:

1. We formalize readability optimization as the simultaneous minimization of three objectives, a readability grade, an edit rate, and a semantic distance, and characterize its solutions as a Pareto front of readability / meaning / edit trade-offs rather than a single rewrite (Section 3).
2. We describe an open, dependency-light benchmark framework that decouples the synonym source, the readability metric, the objective set and the optimizer, so that any metaheuristic can be run against an identical problem instance under identical conditions (Section 4).
3. We conduct a controlled comparison of eight optimizers, random search, a plain GA, three single-solution/swarm metaheuristics (Jaya, Cuckoo Search, TLBO) and three dedicated multi-objective evolutionary algorithms (NSGA-II, GDE3, SMPSO), across a ten-document corpus, under an equal evaluation budget, reporting hypervolume, trade-off quality and a Friedman significance test (Section 6).
4. We show, through an objective-count ablation, that adding an explicit semantic objective removes almost all of the meaning drift and unnecessary editing incurred by readability-only search at negligible cost in readability gain (Section 6).

Our central finding is that the choice of *search paradigm* matters as much as the choice of lexical resource: true multi-objective optimizers recover the whole trade-off curve, including the safe, low-edit region that a practitioner would most likely deploy, whereas scalarizing single-objective metaheuristics concentrate on the aggressive-simplification end and over-edit the text. Among the algorithms tested, GDE3 attains the best Pareto coverage, followed by SMPSO and NSGA-II, and the differences are statistically significant.

The benchmark is deliberately *complementary* to LLM-based simplification. It is training-free, deterministic, transparent (every edit is an inspectable word choice with no hallucination risk), and gives explicit control over the readability/meaning/edit trade-off. As the evaluation literature has repeatedly argued, the field lacks strong, reproducible, controllable baselines against which learned systems can be fairly measured [3]; a multi-objective optimization benchmark is exactly such a baseline.

2. Related work

2.1. Readability metrics

Quantitative readability formulas estimate the difficulty of a text from surface features such as sentence length and word length. The Flesch Reading Ease score and the derived Flesch–Kincaid Grade Level remain the most widely used [4, 5], alongside SMOG [6], the Coleman–Liau index [7], the Automated Readability Index [8], the Gunning Fog index [9] and Dale–Chall [10]. These formulas are non-differentiable functions of the text and are cheap to evaluate, which makes them natural fitness functions for black-box optimization but unsuitable for gradient-based training. We use them as objectives to be optimized directly, following [2].

2.2. Automatic text simplification

Text simplification has been considered one dimension of data quality [11]. Lexical simplification is classically decomposed into complex-word identification, substitute generation, selection and ranking; the readability optimizer studied here can be seen as a global, metric-driven instantiation of the last three stages, in which the selection of substitutions across the whole document is decided jointly by a search algorithm. Modern systems increasingly rely on neural sequence-to-sequence models and, most recently, LLMs, which are competitive with trained systems in zero- and few-shot settings [12]. There are numerous application fields such as question-answering [13], text processing [14] or even software engineering [15, 16]. Such models optimize readability only *indirectly*, through prompting or fine-tuning, and provide no explicit, tunable handle on the amount of meaning drift or editing they introduce. Controllable simplification systems expose coarse control tokens (e.g. target length or lexical complexity), but still do not return the *set* of Pareto-optimal alternatives that an explicit multi-objective optimizer produces.

2.3. Evaluation of simplification

Evaluation remains a well-known weak point of the field. Reference-based metrics such as SARI and the associated TurkCorpus [17] or multi-reference datasets such as ASSET [18] have improved comparability, yet automatic metrics correlate only weakly with human judgements of simplicity, meaning preservation and fluency [3]. A recurring recommendation is to report simplicity, meaning preservation and edit magnitude as *separate* axes rather than a single number. Our multi-objective formulation follows this recommendation by construction: readability, semantic distance and edit rate are kept as distinct objectives throughout.

2.4. Multi-objective metaheuristics and evolutionary computation for text

Multi-objective evolutionary algorithms return a set of non-dominated solutions approximating the Pareto front [19]. We evaluate three canonical representatives: NSGA-II, which combines fast non-dominated sorting with crowding distance and elitism [20]; GDE3, a differential-evolution based multi-objective optimizer [21]. We contrast them with three parameter-light single-objective metaheuristics that have become popular baselines, Jaya [22], Cuckoo Search [23] and Teaching–Learning-Based Optimization (TLBO) [24], applied to a scalarized objective, plus a plain GA and uniform random search. Quality is assessed with the hypervolume indicator, which jointly captures convergence and diversity and does not require the true front to be known [25]. Evolutionary computation has a long history in NLP [26], and the recent use of evolutionary search over natural-language prompts for LLMs shows renewed interest in metaheuristics that operate on discrete linguistic objects.

3. Problem formulation

Encoding. Let the input document T_0 be a sequence of tokens $w_1, \dots, w_L$. A token is a *candidate* for replacement if it is at least four characters long, is not capitalized (a cheap proper-noun guard) and has at least one synonym in the active lexical resource. For each candidate w_i the resource returns an ordered list of synonyms $S_i = (s_{i,1}, \dots, s_{i,k_i})$. A solution is a real vector $\mathbf{x} \in [-4, 4]^L$, one gene per token. Gene x_i is decoded by rounding: if w_i is a candidate and $\lceil x_i \rceil \geq 1$, the token is replaced by $s_{i, \min(\lceil x_i \rceil, k_i)}$; otherwise the original token is kept. This continuous relaxation lets every optimizer, evolutionary, swarm or differential, search the same box-constrained space, while the rounding and clamping recover a discrete edit. We write $T(\mathbf{x})$ for the reconstructed text.

Objectives. We minimize three objectives simultaneously:

$$f_{\text{read}}(\mathbf{x}) = R(T(\mathbf{x})), \quad (1)$$

$$f_{\text{edit}}(\mathbf{x}) = \frac{|\{i : \text{candidate } w_i \text{ changed}\}|}{\#\text{candidates}}, \quad (2)$$

$$f_{\text{sem}}(\mathbf{x}) = d(T_0, T(\mathbf{x})), \quad (3)$$

where $R(\cdot)$ is a readability grade (Flesch–Kincaid Grade Level, FKGL, in our experiments; lower is easier), f_{edit} is the fraction of candidate tokens actually modified, and $d(\cdot, \cdot)$ is a semantic distance between the original and the rewrite. For metrics on which *higher* means easier (e.g. Flesch Reading Ease) the sign of f_{read} is flipped so that minimization always pushes toward easier text. As the semantic distance we use, in the reproducible experiments below, a dependency-free Jaccard distance over token sets, $d(T_0, T) = 1 - |A \cap B|/|A \cup B|$ with A, B the word sets of T_0 and T ; the framework also supports Word Mover’s Distance [27] over Word2Vec embeddings [28], which was the meaning measure used by [2].

Pareto optimality. A solution $\mathbf{x}$ dominates $\mathbf{x}'$ if it is no worse on every objective and strictly better on at least one. The set of non-dominated solutions is the Pareto front; each of its members is a different, defensible compromise between readability, meaning and editing. A single-objective solver that minimizes the sum $f_{\text{read}} + f_{\text{edit}} + f_{\text{sem}}$ returns a single point of this front, and because the readability grade is numerically much larger than the two normalized $[0, 1]$ objectives, that point sits at the readability-dominated extreme, as our results confirm.

4. The benchmark framework

Our experiments are built on ORUGA, an open framework for readability optimization and benchmarking that we extend from a single-objective GA [2] into a composable, multi-objective engine. The pipeline is

text+provider $\rightarrow$ encoding $\rightarrow$ problem (objectives) $\rightarrow$ optimizer $\rightarrow$ report (before/after + Pareto front),

and every stage is selectable independently:

- **Synonym providers.** A deterministic in-memory lexicon (used here for full reproducibility), WordNet [29], Word2Vec nearest neighbours [28], and a web thesaurus.

- **Readability metrics.** FKGL, Flesch Reading Ease, SMOG, Coleman–Liau, ARI and Gunning Fog, implemented in dependency-free Python and cross-validated against an independent library.
- **Objectives.** Readability alone, or with an edit-rate objective, or with an additional semantic-distance objective (Jaccard or Word Mover’s Distance).
- **Optimizers.** Random search and a GA (pure Python); Jaya, Cuckoo Search and TLBO (NumPy); and NSGA-II, GDE3 and SMPSO. Every optimizer consumes the same problem object and archives all evaluations, so a Pareto front can be reported even for the scalarizing methods.

Because the synonym source, metric and objective set are fixed by configuration, all optimizers face an *identical* search space and fitness landscape; the only thing that varies in the comparison is the search strategy.

5. Experimental setup

Corpus. We use a corpus of ten encyclopaedic passages (Table 1), on average 124.7 words long, spanning general-science, history, geography, sports and technology topics. Their mean Flesch–Kincaid Grade Level is 13.56, i.e. roughly college level, representative of text that would benefit from simplification.

Lexicon. To make the optimizer comparison fully reproducible and free of any network or model dependency, all runs draw candidate substitutions from a single curated simplification lexicon of 228 entries mapping complex words to ordered, meaning-preserving simpler alternatives (e.g. *ramifications*→*results/effects*, *visualization*→*charts*, *approximately*→*about*). Using one fixed lexicon for every algorithm is a deliberate design choice: it isolates the behaviour of the *optimizer* from the noise of the lexical resource, which is what the benchmark is meant to measure. Under this lexicon each document exposes between 14 and 38 replaceable tokens (mean 23.3).

Protocol. Every optimizer is given an equal budget of approximately 1000 fitness evaluations and a population of 20 (where applicable); single-solution methods are matched on total evaluations. The three objectives are FKGL, edit rate and Jaccard semantic distance. Each (document, optimizer) pair is run for three random seeds, giving $10 \times 8 \times 3 = 240$ runs. For the objective-count ablation we additionally run GA and NSGA-II under the single-objective (SO), bi-objective (BI: readability + edit rate) and tri-objective (TRI) settings for two seeds. All experiments were executed on a 2-core machine; wall-clock times are reported per run.

Quality measures. For each run we report (i) the largest FKGL reduction found on the front; (ii) a *knee* compromise solution, obtained by min–max normalizing each objective within the front and selecting the point closest to the ideal corner, together with its edit rate and semantic distance; and (iii) the hypervolume of the front. Hypervolume is computed after normalizing every objective to $[0, 1]$ using the per-document range observed across all runs, with reference point $(1.1, 1.1, 1.1)$; larger is better [30]. We assess statistical significance with a Friedman test on the per-document mean hypervolume across the eight optimizers.

Table 1: The ten-document benchmark corpus and its baseline Flesch–Kincaid Grade Level (FKGL). Passages are encyclopaedic excerpts.

#	Topic	Words	FKGL ₀
1	The sea / oceanography	106	14.94
2	History of Austria	118	13.71
3	Mediterranean Sea	125	13.96
4	Real Madrid	104	11.58
5	Programming languages	117	13.30
6	Niagara Falls	156	9.87
7	Big data	115	14.97
8	Tumacácori mission	150	15.38
9	2020 Belgian Super Cup	123	18.92
10	Mapagala fortress	133	8.99
Mean		124.7	13.56

6. Results

6.1. Many-objective optimizer comparison

Table 2 reports the tri-objective comparison. The three dedicated multi-objective optimizers dominate on hypervolume: GDE3 achieves the best front quality (0.639), followed by SMPSO (0.607) and NSGA-II (0.596), clearly ahead of TLBO (0.556), random search (0.544), the plain GA (0.511), Jaya (0.494) and Cuckoo Search (0.411). A Friedman test on the per-document mean hypervolume rejects the null hypothesis of equal performance ($\chi^2 = 64.9$, $p = 1.6 \times 10^{-11}$, eight optimizers over ten documents), and the mean ranks (Table 3) place GDE3 first (1.2), SMPSO second (2.0) and NSGA-II third (3.0).

The compromise-solution columns explain *why*. The multi-objective optimizers return knee solutions that improve readability while editing only a handful of words and drifting negligibly in meaning (SMPSO: edit rate 0.009, semantic distance 0.003; GDE3: 0.036/0.014; NSGA-II: 0.068/0.028). The scalarizing metaheuristics instead push readability hard, Jaya attains the single largest FKGL reduction (2.99 grade levels on average), but pay for it with three to twenty times more editing (Jaya 0.173, GA 0.175, Cuckoo 0.190) and correspondingly larger meaning drift. In other words, minimizing a summed objective drives the search to the readability-dominated corner of the front and never surfaces the conservative, low-risk rewrites that a practitioner would most likely deploy.

Figure 1 makes the mechanism visible on a single document. The fronts of GDE3, SMPSO and NSGA-II extend from the low-edit region (barely any changes, readability close to the original) all the way to aggressive rewrites, giving the user a continuum of choices. The GA and Jaya fronts, by contrast, only populate the high-edit end of the curve, they never offer a solution that edits fewer than about a fifth of the candidate words, precisely because scalarization makes those conservative solutions look uncompetitive.

6.2. Does the semantic objective matter? An ablation

Table 4 isolates the effect of adding objectives, using GA and NSGA-II. With readability as the *only* objective, both algorithms achieve the largest FKGL improvement, but the resulting rewrite is unconstrained, and adding an edit-rate objective alone still leaves a very high edit rate (0.70

Table 2: Tri-objective comparison (FKGL, edit rate, Jaccard semantic distance) over ten documents $\times$ three seeds, equal budget of ≈ 1000 evaluations. Hypervolume (HV) and FKGL improvement are mean $\pm$ s.d.; the compromise columns describe the knee solution of the front. Higher HV and FKGL improvement are better; lower edit rate and semantic distance are better. Best in bold.

Optimizer	Front quality		Compromise (knee) solution				Front	ms/run
	HV	FKGL impr.	FKGL	Edit rate	Sem. dist.			
Random	0.544±.052	2.26±0.98	13.11	0.093	0.043	23.6	1086	
GA	0.511±.071	2.69±1.30	12.79	0.175	0.084	36.3	1078	
Jaya	0.494±.079	2.99±1.51	12.89	0.173	0.080	31.7	1193	
Cuckoo	0.411±.041	1.89±0.91	12.88	0.190	0.080	15.5	1119	
TLBO	0.556±.082	2.75±1.31	12.83	0.151	0.070	31.2	1131	
NSGA-II	0.596±.061	2.21±1.04	13.15	0.068	0.028	14.0	848	
GDE3	0.639±.063	2.48±1.15	13.28	0.036	0.014	16.8	779	
SMPSO	0.607±.046	2.49±1.17	13.51	0.009	0.003	18.2	1004	

Table 3: Mean Friedman rank on per-document hypervolume (1 = best). $\chi^2 = 64.9$, $p = 1.6 \times 10^{-11}$.

	GDE3	SMPSO	NSGA-II	TLBO	Random	GA	Jaya	Cuckoo
Mean rank	1.2	2.0	3.0	4.3	4.7	6.3	6.5	8.0

for GA, 0.54 for NSGA-II) because nothing rewards keeping the original wording. Introducing the semantic-distance objective changes the picture completely: for NSGA-II the knee solution’s edit rate collapses from 0.54 to 0.06 and its semantic distance to 0.02, at a cost of essentially zero readability improvement (2.32 $\rightarrow$ 2.34). The explicit meaning objective is therefore almost “free”: it removes needless edits and meaning drift without sacrificing readability gains, an argument for keeping simplicity, meaning and edit magnitude as separate objectives.

6.3. Convergence

Figure 2 shows single-objective FKGL convergence on document #7. Jaya converges fastest and deepest (12.23 $\rightarrow$ 9.99), with the GA and TLBO close behind, whereas Cuckoo Search stalls almost immediately and random search plateaus early. This is consistent with the tri-objective results: the aggressive single-objective descent that makes Jaya and the GA attractive when only readability matters is exactly what makes them over-edit when meaning must be preserved.

6.4. Qualitative example

Table 5 shows a rewrite selected from a GDE3 front (document #7) under a modest edit budget. Readability improves from FKGL 14.97 to 12.50 while only 27% of candidate words change and the semantic distance stays low (0.153). Every edit is an inspectable, meaning-preserving substitution (*statistical* $\rightarrow$ *math*, *attributes* $\rightarrow$ *traits*, *visualization* $\rightarrow$ *charts*, *velocity* $\rightarrow$ *speed*); there is no hallucination, and the document structure is untouched. The occasional imperfect choice (*analysis* $\rightarrow$ *review*) reflects the fixed lexicon rather than the optimizer, and would be avoided with a richer synonym resource.

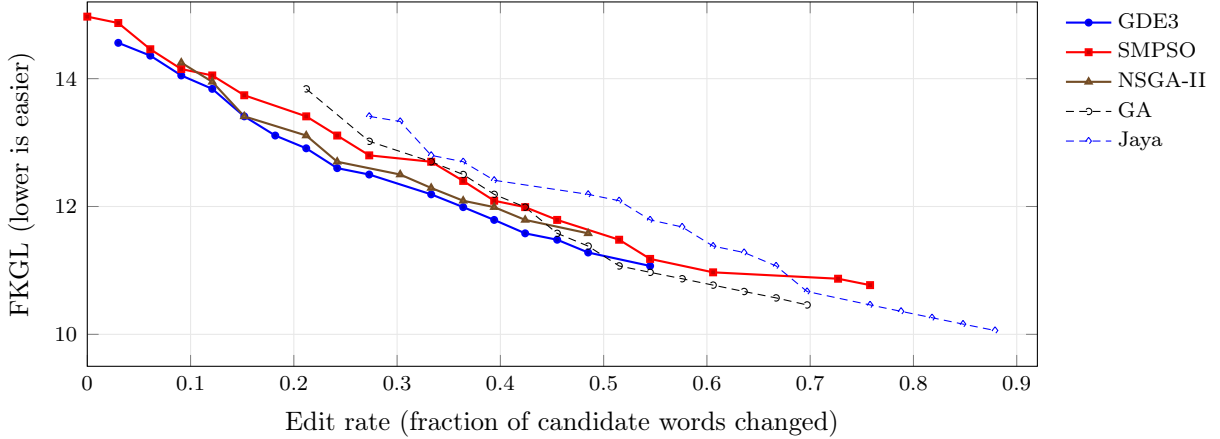


Figure 1: Readability/edit trade-off fronts (document #7, “Big data”, projected onto the FKGL vs. edit-rate plane). Multi-objective optimizers (solid) cover the whole front, including the low-edit region; scalarizing metaheuristics (dashed) only reach the aggressive, high-edit end.

Table 4: Objective-count ablation (ten documents $\times$ two seeds). SO: readability only; BI: + edit rate; TRI: + semantic distance. Edit rate and semantic distance are for the compromise solution.

Optimizer	Setting	FKGL impr.	Edit rate	Sem. dist.
GA	SO	2.84	—	—
GA	BI	2.84	0.699	—
GA	TRI	2.81	0.160	0.073
NSGA-II	SO	2.77	—	—
NSGA-II	BI	2.32	0.542	—
NSGA-II	TRI	2.34	0.062	0.020

7. Discussion

7.1. Lessons learnt

Three practical messages emerge. First, *the search paradigm is a first-order design decision*. On identical problem instances, with identical lexical resources and an identical evaluation budget, dedicated multi-objective optimizers produced markedly better and more useful trade-off sets than scalarizing single-objective metaheuristics; the gap is statistically significant and stable across documents. Among the tested algorithms GDE3 is the recommended default, with SMPSO and NSGA-II close behind. Second, *meaning preservation should be an explicit objective, not a regularizer folded into a single score*. The ablation shows that adding a semantic objective all but eliminates unnecessary editing and drift at negligible readability cost. Third, *returning a Pareto front, not a point, is what makes the method deployable*: different audiences and documents call for different points on the readability/meaning/edit curve, and only the multi-objective optimizers expose that curve.

These properties position the benchmark as a complement to LLM-based simplification rather than a competitor. LLMs excel at fluent rewriting but provide neither an explicit trade-off surface nor reproducibility guarantees, and they can hallucinate content. The optimization approach is training-free, deterministic, transparent and controllable, and because it scores every candidate

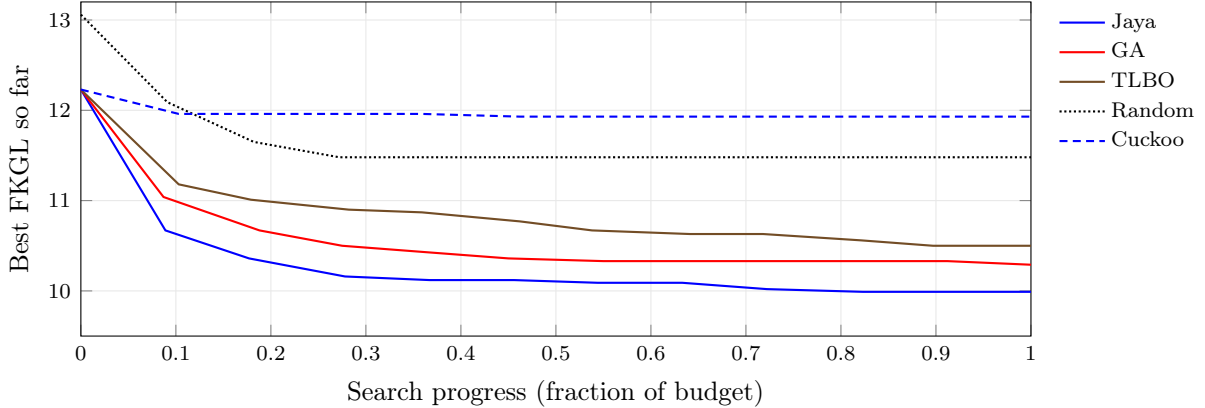


Figure 2: Single-objective FKGL convergence (document #7, mean of three seeds, x normalized to the shared evaluation budget).

Table 5: Example rewrite (document #7, GDE3 front, edit rate 0.27, semantic distance 0.153). Substituted words are underlined.

Original (FKGL 14.97)	Big data refers to data sets that are too large or complex to be dealt with by traditional data-processing application software. Data with many fields (rows) offer greater <i>statistical</i> power, while data with higher complexity (more <i>attributes</i> or columns) may lead to a higher false discovery rate. Big data <i>analysis</i> challenges include capturing data, data storage, data <i>analysis</i> , search, sharing, transfer, <i>visualization</i> , querying, updating, information privacy, and data source. Big data was <i>originally</i> associated with three key concepts volume, variety, and <i>velocity</i> .
Simplified (FKGL 12.50)	Big data refers to data sets that are too large or complex to be dealt with by traditional data-processing application software. Data with many fields (rows) offer greater <u>math</u> power, while data with more complexity (more <u>traits</u> or columns) may lead to a higher false discovery rate. Big data <u>review</u> challenges include capturing data, data storage, data <u>review</u> , search, sharing, transfer, <u>charts</u> , querying, updating, information privacy, and data source. Big data was <u>at first</u> associated with three key concepts volume, variety, and <u>speed</u> .

with the same public readability and meaning measures, it yields auditable baselines against which LLM outputs can be placed. A natural hybrid, which the framework already supports in principle, is to let an LLM propose the synonym candidates and let the multi-objective optimizer choose among them under explicit readability, meaning and edit constraints.

7.2. Threats to validity

Construct validity. We optimize FKGL and a Jaccard semantic distance; these are proxies. FKGL rewards shorter words and sentences and can be gamed, and Jaccard overlap is a coarse meaning measure. The framework mitigates this by supporting six readability formulas and Word Mover’s Distance over embeddings, and by keeping meaning as a separate reported axis; nonetheless, human evaluation of the rewrites is future work. *Internal validity.* Results are averaged over three seeds and a Friedman test controls for chance; the equal-evaluation-budget protocol ensures no optimizer is advantaged by more computation. *External validity.* The corpus is ten encyclopaedic passages and the lexicon is curated and fixed. This maximizes reproducibility and isolates optimizer

behaviour, but absolute readability gains are bounded by the lexicon and will differ with WordNet or embedding-based providers and on other genres. The *relative* ordering of optimizers, which is the object of study, is expected to be more robust than the absolute numbers. *Reproducibility.* All providers, metrics and optimizers are selectable by configuration, all randomness is seeded, and the readability metrics are implemented without external dependencies and cross-validated, so every number in this paper can be regenerated.

8. Conclusion

We recast semantics-preserving readability optimization as a multi-objective problem and built a reproducible benchmark for comparing metaheuristics on the readability/meaning/edit trade-off. Across ten documents and eight optimizers under an equal budget, dedicated multi-objective evolutionary algorithms, GDE3 first, then SMPSO and NSGA-II, significantly outperformed scalarizing single-objective metaheuristics, which systematically over-edit the text to chase readability, and we showed that an explicit semantic objective removes almost all of that over-editing for free. Beyond the specific ranking, the study argues that simplification research should report Pareto fronts over separate simplicity, meaning and edit axes, and should use transparent, controllable optimization baselines alongside learned systems. Future work includes human evaluation of the trade-off solutions, embedding- and LLM-based candidate generation within the same multi-objective search, many-objective formulations that add fluency and grammaticality as further axes, and extension beyond English.

References

- [1] M. Shardlow, A survey of automated text simplification, *International Journal of Advanced Computer Science and Applications* 4 (1) (2014) 58–70. doi:10.14569/SpecialIssue.2014.040109.
- [2] J. Martinez-Gil, Optimizing readability using genetic algorithms, *Knowledge-Based Systems* 284 (2024) 111273. doi:10.1016/j.knosys.2023.111273.
- [3] E. Sulem, O. Abend, A. Rappoport, BLEU is not suitable for the evaluation of text simplification, in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018, pp. 738–744. doi:10.18653/v1/D18-1081.
- [4] R. Flesch, A new readability yardstick, *Journal of Applied Psychology* 32 (3) (1948) 221–233. doi:10.1037/h0057532.
- [5] J. P. Kincaid, R. P. Fishburne, R. L. Rogers, B. S. Chissom, Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel, Tech. Rep. Research Branch Report 8-75, Naval Air Station Memphis, Chief of Naval Technical Training (1975).
- [6] G. H. McLaughlin, SMOG grading: A new readability formula, *Journal of Reading* 12 (8) (1969) 639–646.
- [7] M. Coleman, T. L. Liau, A computer readability formula designed for machine scoring, *Journal of Applied Psychology* 60 (2) (1975) 283–284. doi:10.1037/h0076540.

- [8] E. A. Smith, R. J. Senter, Automated readability index, Tech. Rep. AMRL-TR-66-22, Aerospace Medical Research Laboratories (1967).
- [9] R. Gunning, The Technique of Clear Writing, McGraw-Hill, New York, 1952.
- [10] E. Dale, J. S. Chall, A formula for predicting readability, Educational Research Bulletin 27 (1) (1948) 11–20, 37–54.
- [11] J. Martinez-Gil, Framework to automatically determine the quality of open data catalogs, Expert Systems with Applications 289 (2025) 128379.
- [12] Y. Feng, J. Qiang, Y. Li, Y. Yuan, Y. Zhu, Sentence simplification via large language models, arXiv preprint arXiv:2302.11957 (2023).
- [13] J. Martinez-Gil, A survey on legal question–answering systems, Computer Science Review 48 (2023) 100552.
- [14] J. Martinez-Gil, An overview of textual semantic similarity measures based on web intelligence, Artificial Intelligence Review 42 (4) (2014) 935–943.
- [15] J. Martinez-Gil, Source code clone detection using unsupervised similarity measures, in: International conference on software quality, Springer, 2024, pp. 21–37.
- [16] J. Martinez-Gil, Advanced detection of source code clones via an ensemble of unsupervised similarity measures, in: International Conference on Software Quality, Springer, 2025, pp. 72–90.
- [17] W. Xu, C. Napoles, E. Pavlick, Q. Chen, C. Callison-Burch, Optimizing statistical machine translation for text simplification, Transactions of the Association for Computational Linguistics 4 (2016) 401–415. doi:10.1162/tacl_a_00107.
- [18] F. Alva-Manchego, L. Martin, A. Bordes, C. Scarton, B. Sagot, L. Specia, ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), 2020, pp. 4668–4679. doi:10.18653/v1/2020.acl-main.424.
- [19] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, Chichester, 2001.
- [20] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197. doi:10.1109/4235.996017.
- [21] S. Kukkonen, J. Lampinen, GDE3: The third evolution step of generalized differential evolution, in: Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC), Vol. 1, 2005, pp. 443–450. doi:10.1109/CEC.2005.1554717.
- [22] R. V. Rao, Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems, International Journal of Industrial Engineering Computations 7 (1) (2016) 19–34. doi:10.5267/j.ijiec.2015.8.004.

- [23] X.-S. Yang, S. Deb, Cuckoo search via Lévy flights, in: Proceedings of the World Congress on Nature and Biologically Inspired Computing (NaBIC), 2009, pp. 210–214. doi:10.1109/NaBIC.2009.5393690.
- [24] R. V. Rao, V. J. Savsani, D. P. Vakharia, Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems, *Computer-Aided Design* 43 (3) (2011) 303–315. doi:10.1016/j.cad.2010.12.015.
- [25] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach, *IEEE Transactions on Evolutionary Computation* 3 (4) (1999) 257–271. doi:10.1109/4235.797969.
- [26] J. Martinez-Gil, A comprehensive review of stacking methods for semantic similarity measurement, *Machine Learning with Applications* 10 (2022) 100423.
- [27] M. J. Kusner, Y. Sun, N. I. Kolkin, K. Q. Weinberger, From word embeddings to document distances, in: Proceedings of the 32nd International Conference on Machine Learning (ICML), Vol. 37, 2015, pp. 957–966.
- [28] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: Workshop Track Proceedings of the 1st International Conference on Learning Representations (ICLR), 2013.
- [29] G. A. Miller, WordNet: A lexical database for English, *Communications of the ACM* 38 (11) (1995) 39–41. doi:10.1145/219717.219748.
- [30] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, V. Grunert da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, *IEEE Transactions on Evolutionary Computation* 7 (2) (2003) 117–132. doi:10.1109/TEVC.2003.810758.