Biased Mutation Operators for Subgraph-Selection Problems

Günther R. Raidl, Gabriele Koller, and Bryant A. Julstrom

***Abstract*— Many graph problems seek subgraphs of minimum weight that satisfy a set of constraints. Examples include the minimum spanning tree problem (MSTP), the degree-constrained minimum spanning tree problem (d-MSTP), and the traveling salesman problem (TSP). Low-weight edges predominate in optimum solutions to such problems, and the performance of evolutionary algorithms (EAs) is often improved by biasing variation operators to favor these edges. We investigate the impact of biased edge-exchange mutation. In a large-scale empirical investigation on Euclidean and uniform random instances, we describe the distributions of edges in optimum solutions of the MSTP, the d-MSTP, and the TSP in terms of the edges' weight-based ranks. We approximate these distributions by exponential functions and derive approximately optimal probabilities for selecting edges to be incorporated into candidate solutions during mutation. A theoretical analysis of the expected running time of a (1+1)-EA on non-degenerate instances of the MSTP shows that, when using the derived probabilities for edge-selection in mutation, the (1+1)-EA is asymptotically as fast as a classical implementation of Kruskal's minimum spanning tree algorithm. In experiments on the MSTP, d-MSTP, and the TSP, we compare the new edge-selection strategy to four alternative methods. The results of a (1+1)-EA on instances of the MSTP support the theory and indicate that the new strategy is superior to the other methods in practice. On instances of the d-MSTP, a more sophisticated EA with a larger population and unbiased recombination performs better with the new biased mutation than with alternate mutations. On the TSP, the advantages of weight-biased mutation are generally smaller, because the insertion of a specific new edge into a tour requires the insertion of a second, dependent edge as well. Although we considered Euclidean and uniform random instances only, we conjecture that the same biasing towards low-weight edges also works well on other instance classes structured in different ways.**

***Index Terms*— Biased operators, mutation, graph problems, minimum spanning tree problems, traveling salesman problem**

I. Introduction

An undirected graph $G = (V, E)$ consists of a non-empty set V of vertices and a set E of unordered pairs of vertices, called edges. In a weighted undirected graph, a function $w : E \to \mathbb{R}^+$ associates a numerical weight with each edge in E. Many problems on graphs seek a subset S of G's edges that satisfies a set of constraints and has minimum total weight $w(S) = \sum_{e \in S} w(e)$ over all such subsets. The constraints that S must satisfy characterize each problem, as in these examples:

- S is a Hamiltonian tour, a cycle that visits each vertex in G exactly once (the traveling salesman problem; TSP).
- S is a path in G that connects two specified vertices (the shortest path problem).
- S is a spanning tree (the unconstrained minimum spanning tree problem; MSTP).
- S is a spanning tree in which the number of edges incident on each vertex does not exceed a bound $d > 1$ (the degree-constrained minimum spanning tree problem; d-MSTP) [1], [2].
- S is a spanning tree with at least L leaves (the leaf-constrained minimum spanning tree problem) [3], [4].
- S augments a given subgraph so that the resulting network is biconnected (the biconnectivity augmentation problem) [5].
- S is a Steiner tree that connects a specified subset of G's vertices.

Some of these problems, such as the unconstrained MSTP and the identification of a shortest path between two vertices, can be solved to optimality in polynomial time. Most, including the remaining problems listed above, are NP-hard, so it is unlikely that there can be polynomial-time algorithms that will in general solve them exactly. In these cases, (meta-)heuristics, including evolutionary algorithms (EAs), are often useful.

It is not surprising—and the following section verifies—that low-weight edges predominate in solutions to problems like these that seek constrained low-weight subgraphs. Thus, any heuristic that builds candidate solutions to such problems should favor edges of lower weight. Evolutionary algorithms can apply this observation to constructing the solutions in their initial populations and to their recombination and mutation operators, which construct new solutions from existing ones.

Several researchers have examined such mechanisms [6], [7], [8], [9]. Among them, Julstrom and Raidl studied weight-biased crossover operators in EAs for the TSP and the d-MSTP on complete graphs [10]; favoring low-weight edges improved the performance of these algorithms. The present authors investigated weight-biased mutation in these EAs and derived probabilities for selecting edges that minimize the expected time to include edges of optimum tours and trees [11].

This article extends in several ways our work on biased mutation in EAs for subset-selection problems on complete graphs. The next section investigates empirically the distributions of edges in optimum solutions of the MSTP, the d-MSTP, and the TSP, in terms of the edges' weight-based ranks. For all three problems, the probability that an edge of rank r appears in an optimum solution can be closely

G. R. Raidl and G. Koller are with the Institute of Computer Graphics and Algorithms, Vienna University of Technology, 1040 Vienna, Austria (e-mail: {raidl, koller}@ads.tuwien.ac.at).

B. A. Julstrom is with the Department of Computer Science, St. Cloud State University, St. Cloud, MN 56301 USA (e-mail: julstrom@stcloudstate.edu).

approximated by an exponential function of r. Section III analytically approximates optimal edge-selection probabilities for typical instances of the three problems.

Section IV analyzes expected running times for a (1+1)-EA using several different edge-selection strategies on the unconstrained MSTP. We show that, when the EA uses the approximately optimal mutation scheme, its expected running time on a non-degenerate instance of the MSTP is asymptotically not worse than the time of a classical implementation of Kruskal's well-known minimum-spanning-tree algorithm [12]. Section V describes a variety of other strategies with which mutation may select edges for inclusion in new solutions, and Sect. VI describes experiments with the (1+1)-EA for the MSTP that confirm the theoretical results.

Sections VII and VIII compare the edge-selection strategies listed in Sect. V in more sophisticated EAs for the d-MSTP and the TSP, respectively. These EAs use larger populations and problem-specific recombination operators in addition to edge-exchange-based mutation. On the d-MSTP, theoretically approximately optimal edge-selection increases the probability of finding optimum solutions and reduces the number of iterations usually needed. On the TSP, the advantages of weight-biased approaches are generally smaller because mutation that (heuristically) introduces one edge into a tour necessarily introduces a second edge as well.

II. Distributions of Edges in Optimum Solutions to the MSTP, d-MSTP, and TSP

It is intuitively reasonable that optimum low-weight trees, tours, and other subgraphs in weighted graphs should contain high proportions of low-weight edges. We confirm and quantify this observation for the computationally easy MSTP and the NP-hard d-MSTP and TSP on complete graphs $G = (V, E)$. Let $n = |V|$ be the number of vertices in G, $m = |E| = n \cdot (n-1)/2$ be the number of edges, and S be the set of edges in a solution, so that $|S| = n - 1$ for the MSTP and d-MSTP and $|S| = n$ for the TSP.

Our sample problem instances are of two kinds, uniform and Euclidean. In the uniform instances, edge weights are integers chosen at random and independently from the interval $[1, 10\,000]$. In the Euclidean instances, vertices are distinct points in the plane whose coordinates are random integers from the interval $[1, 10\,000]$, and edge weights are equal to the Euclidean distances between the points. We generated $1\,000$ instances of each type with $n =$ 20, 50, 100, 200, 500, and $1\,000$ vertices. For the d-MSTP, the degree bound d was set to three and five in turn on the uniform instances. For the Euclidean instances, only the case $d = 3$ was considered since for such instances there always exists an unconstrained minimum spanning tree (MST) of degree no more than five.

Kruskal's algorithm identified unconstrained MSTs for all $6\,000$ instances. Exact algorithms for the d-MSTP and the TSP become infeasible on larger graphs, so on these problems, only the instances with $n = 100$ or fewer vertices were considered. The corresponding 3-MSTP, 5-MSTP, and TSP instances were solved to optimality by branch-and-cut algorithms implemented using the ABACUS environment [13] and CPLEX 8.1 as a linear programming solver.

In each instance, sorting the edges into ascending order of their weights assigns each a rank r, $1 \le r \le m$; ties are broken arbitrarily. We consider edges' ranks because, unlike edges' weights, they can be compared across instances. Figure 1 plots the empirical probabilities—the relative frequencies—$p_E(r)$ with which an edge of rank r appears in the optimum solution. Only the portions of the curves where $p_E(r)$ is visibly larger than zero are shown. Note that the probabilities $p_E(r)$ sum to $|S|$:

$$\sum_{r=1}^{m} p_E(r) = |S| \, . \tag{1}$$

As expected, optimum solutions consist mostly of edges of low rank; that is, of low weight. Moreover, for each kind of problem and each fraction $k \in (0, (n-1)/2]$, the probability $p_E(\lceil kn \rceil)$ that the edge of rank $\lceil kn \rceil$ appears in the optimum solution is approximately constant across all the problem sizes.

Table I documents further properties. For each problem type and size, it lists the number R of lowest-weight edges among which γ = 50, 90, and 99 percent of the optimum solutions' edges are found; i.e., R is the smallest rank for which the cumulated probabilities of the edges of that or lower rank sum to at least $\gamma \cdot |S|$:

$$R = \min \left\{ t : \sum_{r=1}^{t} p_E(r) \ge \gamma \cdot |S| \right\} . \tag{2}$$

Table I also lists the proportions $k = R/n$ of low-weight edges in optimum solutions. For each kind of problem and each value of γ, these values are nearly constant.

An effective heuristic mutation operator in EAs for graph problems like those considered here should introduce edges depending on the probabilities with which they appear in optimum solutions. Toward that end, we approximate the empirical distributions $p_E(r)$ with closed-form expressions $p_A(r)$.

In the graphs in Figure 1, $p_E(r)$ decreases approximately exponentially as r grows, particularly in the Euclidean instances. Therefore, let

$$p_A(r) = a^r \quad \text{with } 0 < a < 1 \, . \tag{3}$$

The base a should be chosen so that

$$\sum_{r=1}^{m} p_A(r) = \sum_{r=1}^{m} a^r = \frac{a - a^{m+1}}{1-a} = |S| \, . \tag{4}$$

The term a^{m+1} is negligible for problems of even moderate size, so we neglect it to obtain

$$\sum_{r=1}^{m} p_A(r) \approx \frac{a}{1-a} \quad \Rightarrow \quad a \approx \frac{|S|}{|S|+1} \, . \tag{5}$$

Figure 2 plots $p_A(r) = a^r$ with $a = |S|/(|S|+1)$ for the 3-MSTP instances with 100 vertices. The graph illustrates that $p_A(r)$ approximates the empirical probabilities $p_E(r)$ with high accuracy. To quantify this accuracy, we calculate the relative mean-square error

$$RMSE = \frac{\sum_{r=1}^{m} (p_A(r) - p_E(r))^2}{\sum_{r=1}^{m} p_E(r)^2} . \tag{6}$$

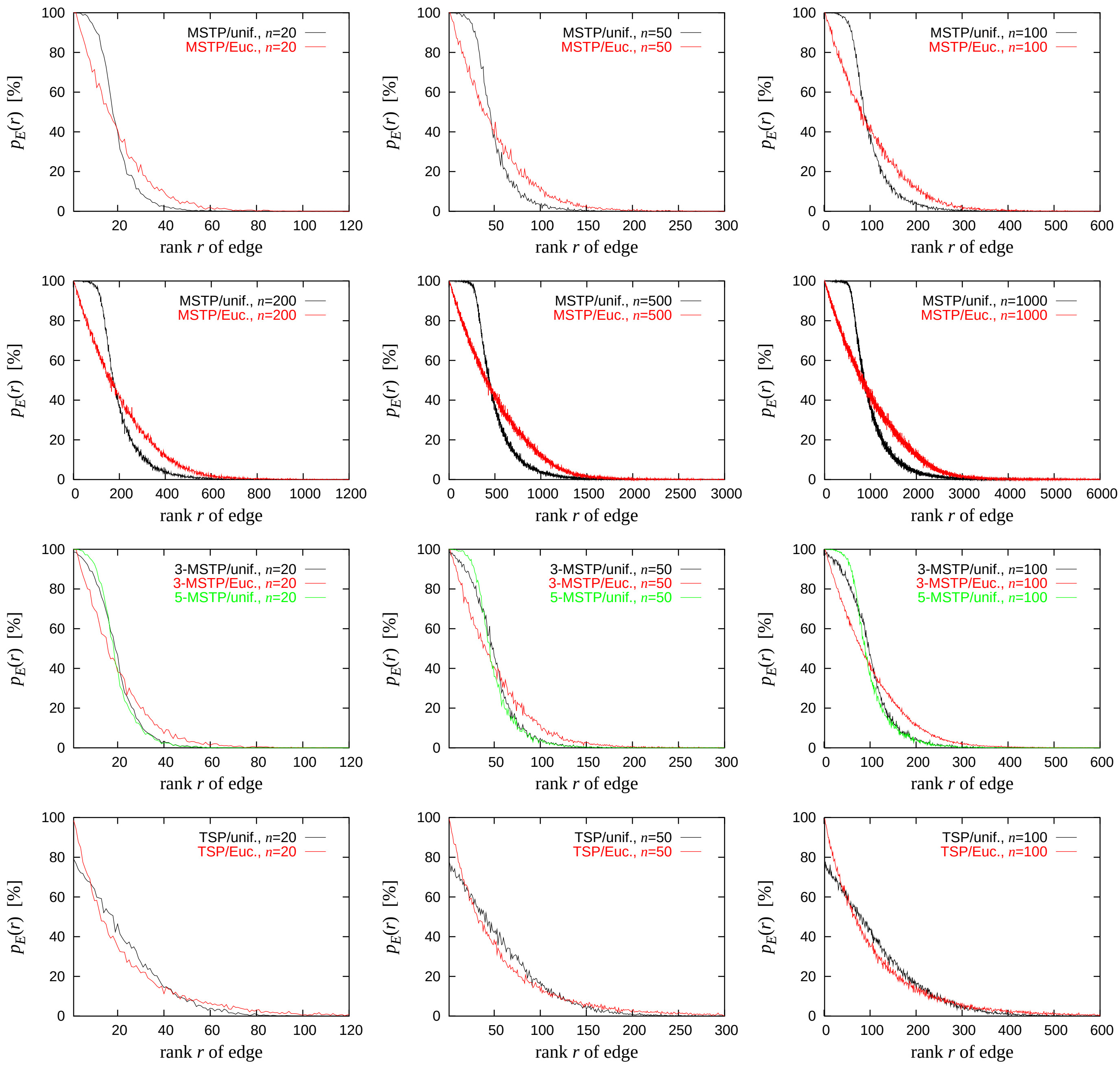


Fig. 1. The empirical probability $p_E(r)$ that an edge appears in an optimum solution as a function of its rank, for the MSTP, 3-MSTP, 5-MSTP, and TSP on uniform and Euclidean instances of size n.

The $RMSE$ allows better comparisons of approximation quality across different problem sizes than does the standard mean-square error. The latter never exceeds 1% on any case and decreases rapidly with increasing problem size n.

Table II lists the $RMSEs$ for all the problem types and sizes; they are always less than 9.9%. In general, $p_A(r)$ approximates $p_E(r)$ more accurately on Euclidean instances than on uniform ones. The approximation is most accurate on the TSP, while the largest errors occur on the unconstrained MSTP on uniform graphs. The approximation is conservative in that it typically underestimates the probabilities $p_E(r)$ of low-rank edges and slightly overestimates the probabilities of high-rank edges.

III. Approximating Optimal Edge-Selection Probabilities

Consider a subset-selection problem on a graph G for which $S^* \subset E$ is the unique optimum solution. Uniformly random edge-exchange mutation chooses each edge to include in a solution with probability $1/m$. We apply the analysis of Sect. II to identify non-uniform probabilities, associated with the edges' ranks, that are optimal in the following sense: Over all edges e^* in the optimum solution S^*, the average expected number of edge-selections until e^* is chosen is minimal.

Let $q(r)$ be the probability that an edge-selection scheme chooses the edge e_r whose rank is r ($1 \le r \le m$). The number

TABLE I

NUMBERS R OF LOWEST-WEIGHT EDGES FOR EACH PROBLEM TYPE AND SIZE n AMONG WHICH $\gamma = 50\%$, 90%, AND 99% OF OPTIMUM SOLUTIONS' EDGES ARE FOUND AND FRACTIONS $k = R/n$.

Problem	n	$\gamma = 50\%$ R	k	$\gamma = 90\%$ R	k	$\gamma = 99\%$ R	k
MSTP/unif.	20	10	0.5	23	1.1	40	2.0
	50	25	0.5	59	1.2	109	2.2
	100	50	0.5	120	1.2	224	2.2
	200	101	0.5	242	1.2	458	2.3
	500	251	0.5	606	1.2	1151	2.3
	1000	501	0.5	1216	1.2	2313	2.3
MSTP/Euc.	20	12	0.6	33	1.6	62	3.1
	50	31	0.6	89	1.8	165	3.3
	100	63	0.6	178	1.8	319	3.2
	200	128	0.6	356	1.8	623	3.1
	500	319	0.6	882	1.8	1496	3.0
	1000	642	0.6	1753	1.8	2913	2.9
3-MSTP/unif.	20	11	0.6	24	1.2	41	2.0
	50	27	0.5	63	1.3	113	2.3
	100	54	0.5	126	1.3	228	2.3
3-MSTP/Euc.	20	12	0.6	33	1.6	63	3.1
	50	31	0.6	89	1.8	165	3.3
	100	63	0.6	179	1.8	324	3.2
5-MSTP/unif.	20	10	0.5	23	1.1	40	2.0
	50	25	0.5	60	1.2	110	2.2
	100	51	0.5	120	1.2	223	2.2
5-MSTP/Euc.	20	12	0.6	33	1.6	63	3.1
	50	31	0.6	89	1.8	165	3.3
	100	63	0.6	178	1.8	323	3.2
TSP/unif.	20	16	0.8	41	2.0	67	3.4
	50	40	0.8	107	2.1	183	3.7
	100	80	0.8	217	2.2	373	3.7
TSP/Euc.	20	15	0.8	53	2.6	107	5.3
	50	37	0.7	134	2.7	297	5.9
	100	73	0.7	257	2.6	587	5.9

of selections until e_r is chosen the first time has a geometric distribution with the expected value

$$EX(e_r) = \frac{1}{q(r)} . \tag{7}$$

Let $p(r)$ be the probability that the edge e_r appears in the optimum solution S^*, and let e^* be a specific edge from S^*. The following theorem establishes the probabilities with which edges should be selected from E to achieve a minimum waiting time for selecting e^*.

Theorem 1: The expected number $EX(e^*)$ of edge-selections until a specific edge $e^* \in S^*$ is chosen for the first time is minimized by the edge-selection probabilities

$$q(r) = \frac{\sqrt{p(r)}}{\sum_{i=1}^{m} \sqrt{p(i)}}. \tag{8}$$

Proof: The probability that e^* has rank r is $p(r)/|S|$. The expected number of edge-selections until e^* is chosen for the first time is the weighted sum

$$EX(e^*) = \sum_{r=1}^{m} \frac{p(r)/|S|}{q(r)} = \frac{1}{|S|} \sum_{r=1}^{m} \frac{p(r)}{q(r)} . \tag{9}$$

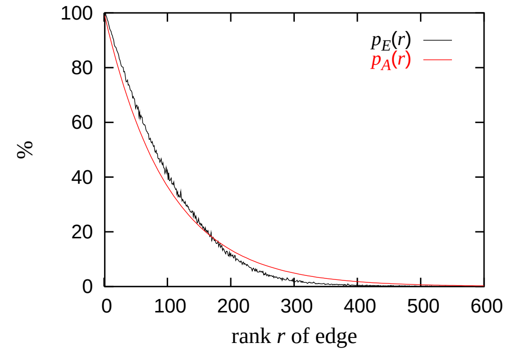


Fig. 2. Approximation of $p_E(r)$ by $p_A(r) = a^r$ for the Euclidean 3-MSTP on $n = 100$ vertices.

TABLE II

RELATIVE MEAN-SQUARE ERRORS WHEN APPROXIMATING EMPIRICAL PROBABILITIES $p_E(r)$ BY $p_A(r) = a^r$ WITH $a = |S|/(|S|+1)$.

Problem	*RMSE* [%] $n = 20$	50	100	200	500	1 000
MSTP/unif.	9.86	9.34	9.26	9.28	9.33	9.33
MSTP/Euc.	1.66	0.98	0.93	0.90	0.98	1.02
3-MSTP/unif.	7.60	6.92	6.87	–	–	–
3-MSTP/Euc.	1.78	0.99	0.84	–	–	–
5-MSTP/unif.	9.42	9.11	9.17	–	–	–
5-MSTP/Euc.	1.79	1.00	0.85	–	–	–
TSP/unif.	1.80	2.31	2.39	–	–	–
TSP/Euc.	0.28	0.25	0.17	–	–	–

Because $\sum_{r=1}^{m} q(r) = 1$, we can replace $q(m)$ by $1 - \sum_{i=1}^{m-1} q(i)$ in (9) and write

$$EX(e^*) = \frac{1}{|S|} \left(\sum_{r=1}^{m-1} \frac{p(r)}{q(r)} + \frac{p(m)}{1 - \sum_{i=1}^{m-1} q(i)} \right) . \tag{10}$$

To identify selection probabilities $q(r)$ that minimize the expectation $EX(e^*)$, we partially differentiate $EX(e^*)$ with respect to each $q(r)$ and set these derivatives equal to zero:

$$\begin{aligned}
\frac{\partial EX(e^*)}{\partial q(1)} &= \frac{1}{|S|} \left(-\frac{p(1)}{q(1)^2} + \frac{p(m)}{(1 - \sum_{i=1}^{m-1} q(i))^2} \right) = 0 \\
\frac{\partial EX(e^*)}{\partial q(2)} &= \frac{1}{|S|} \left(-\frac{p(2)}{q(2)^2} + \frac{p(m)}{(1 - \sum_{i=1}^{m-1} q(i))^2} \right) = 0 \\
&\cdots \\
\frac{\partial EX(e^*)}{\partial q(m-1)} &= \\
&= \frac{1}{|S|} \left(-\frac{p(m-1)}{q(m-1)^2} + \frac{p(m)}{(1 - \sum_{i=1}^{m-1} q(i))^2} \right) = 0
\end{aligned} \tag{11}$$

This system of $m-1$ equations can be simplified to

$$\begin{aligned}
\frac{p(1)}{q(1)^2} = \frac{p(2)}{q(2)^2} = \cdots = \frac{p(m-1)}{q(m-1)^2} &= \\
= \frac{p(m)}{(1 - \sum_{i=1}^{m-1} q(i))^2} &= \frac{p(m)}{q(m)^2} .
\end{aligned} \tag{12}$$

Let $\varphi = p(r)/q(r)^2$. Then

$$q(r) = \sqrt{\frac{p(r)}{\varphi}} \tag{13}$$

and since

$$\sum_{i=1}^{m} q(i) = 1 = \frac{1}{\sqrt{\varphi}} \sum_{i=1}^{m} \sqrt{p(i)}\,, \tag{14}$$

we conclude that

$$\varphi = \left(\sum_{i=1}^{m} \sqrt{p(i)} \right)^2 \quad \text{and} \quad q(r) = \frac{\sqrt{p(r)}}{\sum_{i=1}^{m} \sqrt{p(i)}}\,. \tag{15}$$

■

A. $EX(e^*)$ for Three Edge-Selection Strategies

The following corollary establishes the expected waiting time until selecting an edge from the optimum solution when approximating $p(r)$ as in Sect. II.

Corollary 1: Under the assumption that $p(r)$ is approximated well by $p_A(r) = a^r$ with $a = |S|/(|S|+1)$ according to Eqs. (3) and (5), and when using the corresponding approximately optimal edge-selection probabilities $q_A(r)$, the average expected number of edge-selections until choosing an edge e^* from the optimum solution is

$$EX^*(e^*) \approx \left(\sqrt{|S|} + \sqrt{|S|+1} \right)^2 . \tag{16}$$

Thus, $EX^*(e^*) \approx 4|S|$ and therefore $\Theta(n)$.

Proof: The optimal edge-selection probabilities $q(r)$ from Theorem 1, when substituted into Eq. (9), yield the following expected number of edge-selections:

$$\begin{aligned} EX^*(e^*) &= \frac{1}{|S|} \sum_{r=1}^{m} \frac{p(r)}{\sqrt{p(r)} \Big/ \sum_{i=1}^{m} \sqrt{p(i)}} = \\ &= \frac{1}{|S|} \left(\sum_{r=1}^{m} \sqrt{p(r)} \right)^2 . \end{aligned} \tag{17}$$

Using the approximation $p_A(r) = a^r$ for $p(r)$, we obtain

$$EX^*(e^*) \approx \frac{1}{|S|} \left(\sum_{r=1}^{m} \sqrt{a^r} \right)^2 = \frac{1}{|S|} \left(\frac{\sqrt{a} - a^{(m+1)/2}}{1 - \sqrt{a}} \right)^2 . \tag{18}$$

Since $a^{(m+1)/2}$ is orders of magnitude smaller than $\sqrt{a}$ even for moderate problem sizes, we disregard it. Further, replacing a by $|S|/(|S|+1)$ according to (5), we obtain:

$$\begin{aligned} EX^*(e^*) &\approx \frac{a}{|S|\,(1-\sqrt{a})^2} = \\ &= \frac{1}{(|S|+1) \left(1 - \sqrt{\frac{|S|}{|S|+1}} \right)^2} = \\ &= \left(\sqrt{|S|} + \sqrt{|S|+1} \right)^2 . \end{aligned} \tag{19}$$

■

By Tschebyscheff's inequality [14], when considering the variance of the geometric distribution, the deviation from this expected number of needed edge-selections is less than $\lambda \cdot EX^*(e^*)$ for any $\lambda > 0$ with probability greater than $1 - 1/\lambda^2 = 1 - o(1)$.

Consider the same expected value when edges are selected according to uniform probabilities: for all $r = 1, \ldots, m$, $q_U(r) = 1/m$. Since $\sum_{r=1}^{m} p(r) = |S|$,

$$EX^U(e^*) = \frac{1}{|S|} \sum_{r=1}^{m} \frac{p(r)}{1/m} = \frac{m}{|S|} \sum_{r=1}^{m} p(r) = m\,. \tag{20}$$

Similarly, let edges' probabilities be proportional to $p(r)$: for all $r = 1, \ldots, m$, $q_P(r) = p(r)/|S|$. Then

$$EX^P(e^*) = \frac{1}{|S|} \sum_{r=1}^{m} \frac{p(r)}{p(r)/|S|} = \frac{|S|}{|S|} \sum_{r=1}^{m} 1 = m\,. \tag{21}$$

That is, for both uniform and $p(r)$-proportional probabilities, $EX(e^*) = m$, which is $\Theta(n^2)$, while for the optimal probabilities, $EX^*(e^*)$ is $\Theta(n)$.

B. Approximately Optimal Edge-Selection Probabilities

The preceding observations lead to the following result.

Theorem 2: Under the assumption as in Corollary 1 that $p(r)$ is approximated well by $p_A(r) = a^R$ with $a = |S|/(|S|+1)$, optimal edge-selection probabilities $q(r)$ that minimize $EX(e^*)$ can be closely approximated by

$$q_A(r) \approx \left(\frac{|S|}{|S|+1} \right)^{\frac{r}{2}} \left(\sqrt{\frac{|S|+1}{|S|}} - 1 \right) . \tag{22}$$

Proof: Replacing $p(r)$ by the approximation $p_A(r) = a^r$ in Eq. (8) of Theorem 1 yields a closed-form expression for the optimal edge-selection probabilities $q_A(r)$:

$$\begin{aligned} q_A(r) &= \frac{\sqrt{p_A(r)}}{\sum_{i=1}^{m} \sqrt{p_A(i)}} = \frac{\sqrt{a^r}}{\sum_{i=1}^{m} \sqrt{a^i}} = \frac{\sqrt{a^r}}{\frac{\sqrt{a} - a^{(m+1)/2}}{1-\sqrt{a}}} = \\ &= \frac{(1-\sqrt{a})\, a^{r/2}}{\sqrt{a} - a^{(m+1)/2}} \end{aligned} \tag{23}$$

Again, $a^{(m+1)/2}$ is negligible compared to $\sqrt{a}$, and we ignore it. Again, we replace a with $|S|/(|S|+1)$ according to (5); thus

$$\begin{aligned} q_A(r) &\approx \frac{(1-\sqrt{a})\, a^{r/2}}{\sqrt{a}} = a^{\frac{r}{2}} \left(\frac{1}{\sqrt{a}} - 1 \right) = \\ &= \left(\frac{|S|}{|S|+1} \right)^{\frac{r}{2}} \left(\sqrt{\frac{|S|+1}{|S|}} - 1 \right) . \end{aligned} \tag{24}$$

■

Fig. 3 plots the probabilities $q_A(r)$, $q_U(r)$, and $q_P(r)$ for instances of the 3-MSTP on $n = 100$ vertices.

IV. Expected Running Times of a (1+1)-EA for the MSTP Using Different Edge-Selection Strategies

This section develops the expected times that a (1+1)-EA requires to find an unconstrained MST on a complete graph $G = (V, E)$, using five different edge-selection and replacement strategies in mutation.

Neumann and Wegener [15] investigated simple randomized local search (RLS) and an unbiased (1+1)-EA for the MSTP

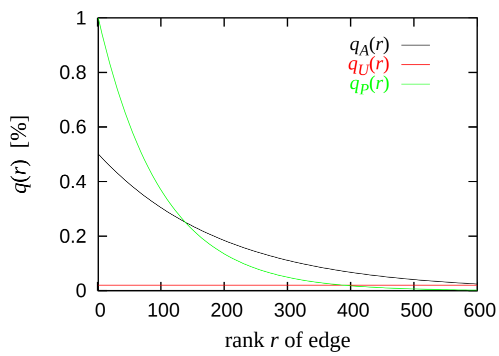


Fig. 3. Edge-selection probabilities $q_A(r)$ (approximately optimal), $q_U(r)$ (uniform), and $q_P(r)$ (proportional to $p_A(r)$) for instances of the 3-MSTP on $n = 100$ vertices.

on general (including incomplete) graphs. Their algorithms encoded candidate subgraphs as bit-strings that indicated the inclusion or exclusion of each edge in G; large penalties in the fitness functions encouraged, first, the formation of connected subgraphs, then their pruning to trees. The neighbor operator of the RLS algorithm flipped one or two randomly chosen bits. The EA's mutation operator flipped each bit independently with probability $1/m$ ($m = |E|$). The expected times of both algorithms to find MSTs were $O(m^2(\log n + \log w_{\max}))$, where $n = |V|$ and $w_{\max}$ is the largest edge weight in G. Neumann and Wegener also showed that MSTP instances exist for which their algorithms' expected times are $\Omega(n^4 \log n)$.

We assume here that all edge weights in each graph are distinct, so that each MST is unique. Furthermore, we restrict our analysis to *non-degenerate MSTs* with the following properties:

1) The diameter of the MST is bounded by $O(\sqrt{n})$; this holds for randomly created spanning trees with overwhelming probability [16], [17].
2) The approximately optimal edge-selection probability $q_A(r)$ of each edge from the MST is not less than the uniform selection probability $1/m$. This assumption is fulfilled by most randomly created spanning trees as long as they are not too small, as our empirical investigations in Sect. II documented. For example, for a MST on $n = 100$ nodes, all edges must have ranks less than $6.41n$, which happens in $\approx 85.3\%$ of all cases. For a MST on $n = 10\,000$ nodes, edge ranks must be less than $15.65n$, which happens with probability $\approx 99.8\%$.

Consider a standard (1+1)-EA for the MSTP that encodes spanning trees directly as edge-sets [8]; these edge-sets can be implemented efficiently by, for example, hash tables. The EA's initial solution is an unbiased random spanning tree, generated via a random walk in the target graph as described by Broder [18]. The expected time of this step is $O(n \log n)$ for almost all graphs, including complete graphs. Each iteration of the EA applies mutation to its current solution to create one offspring. The offspring replaces the incumbent solution if it represents a valid spanning tree of weight no greater than the incumbent's.

In the EA, we compare the following five mutation strategies, in which $T \subseteq E$ is the algorithm's current spanning tree.

- *Naive edge-replacement* (NA): Edges $e \in E - T$ and $e' \in T$ are chosen uniformly at random; e replaces e' in T. This operator may yield solutions that are not spanning trees; such solutions are never accepted.
- *Uniform-uniform edge-replacement* (UU): Again, an edge $e = (u, v) \in E - T$ is chosen for inclusion at random. The edge e' to be removed is randomly chosen from the path in T that connects u and v, so that the offspring is always a spanning tree.
- *Uniform-greedy edge-replacement* (UG): UG-mutation is identical to UU-mutation, except that the edge e' to be removed is always the edge of largest weight on the path from u to v; ties are broken arbitrarily.
- *Biased-uniform edge-replacement* (BU): An edge $e \in E$ is chosen according to the theoretically derived probability $q_A(r)$ from Sect. III. If e is already contained in T, the solution is not modified. Otherwise, an edge e' is chosen at random from the path that connects e's vertices (as in UU) and replaced by e.
- *Biased-greedy edge-replacement* (BG): The biased selection of an edge $e \in E$ from BU is combined with the greedy choice from UG of the edge e' to be removed.

While NA can be implemented so that its time $t_{\text{mut}}^{\text{NA}}$ is $\Theta(1)$, the other methods run in linear time in the worst case, when T is a path. The diameter of a random spanning tree on G, however, is $\Theta(\sqrt{n})$ with overwhelming probability [16], [17], and thus, on non-degenerate instances, these mutation variants are expected to run in times $t_{\text{mut}}^{\text{UU}}$, $t_{\text{mut}}^{\text{UG}}$, $t_{\text{mut}}^{\text{BU}}$, and $t_{\text{mut}}^{\text{BG}}$ that are all $O(\sqrt{n})$.

To identify the expected times the (1+1)-EA, with each mutation operator, requires to find the MST on an average instance of the MSTP, we begin by describing the expected number of edges that the EA's initial tree and the MST have in common.

Lemma 1: Let G be a complete weighted graph. The number of edges in which a random spanning tree on G and a specified target tree differ is at least $2n/3$ with probability $1 - o(1)$.

Proof: Each edge in G appears in a random spanning tree with probability

$$\frac{n-1}{n(n-1)/2} = \frac{2}{n}.$$

Of the $n - 1$ edges in the target spanning tree, the expected number that also appear in a random tree is then

$$(n-1)\frac{2}{n} = 2\frac{n-1}{n}, \quad \text{and} \quad \lim_{n\to\infty} 2\frac{n-1}{n} = 2. \qquad (25)$$

Thus the expected number of edges in which a random spanning tree differs from the target is $(n-1) - (2 - o(1)) = n - 3 + o(1)$, and by Markov's inequality [14], the probability that there are at least $2n/3$ such edges is at least $1 - (6 - o(1))/n = 1 - o(1)$ for large n. ■

To derive the total expected running time, we make use of the following observation based on two results of Neumann and Wegener [15].

Lemma 2: For a non-minimum spanning tree T of weight $w(T)$ and the MST T^* of weight $w(T^*)$, there always exists a set of $n-1$ *good* edge-exchanges in T such that the average weight decrease of these edge-exchanges is at least $(w(T) - w(T^*))/(n-1)$.

Proof: The proof is constructive. Let $z = |T^* - T|$. There exists a bijection $\alpha : T^* - T \rightarrow T - T^*$ such that $\alpha(e)$ lies on the cycle created by including e in T and the weight of $\alpha(e)$ is not less than the weight of e [19]. The total weight decrease when all z edge-exchanges described by the bijection α are applied to T is $w(T) - w(T^*)$.

To make the edge-exchanges independent of the number of edges in T, extend the set of z edge-exchanges that α specifies with $n - z - 1$ dummy exchanges specified by the bijection $\alpha'(e) = e$, $\forall e \in T \cap T^*$. These exchanges replace an edge that is in both T and T^* by itself and so do not modify T and $w(T)$. From α and α' together, we obtain $n-1$ *good* edge-exchanges with an average weight decrease of at least $(w(T) - w(T^*))/(n-1)$. ■

The next result establishes an upper bound on the expected number of good edge-exchanges we must perform.

Lemma 3: The expected number of good edge-exchanges required to transform a random spanning tree into the non-degenerate MST is $O(n \log n)$.

Proof: The proof imitates that of Theorem 2 in [15]. It begins by replacing each edge weight $w(e)$ in $G = (V, E)$ with the edge's rank $r(e)$. Note that the MST based on the ranks will also be the MST based on the original weights. Assume that each of the $n-1$ good edge-exchanges is equally likely. Let $r(T) = \sum_{e \in T} r(e)$ and $r(T^*) = \sum_{e \in T^*} r(e)$. A good edge-exchange decreases the difference $r(T) - r(T^*)$ on average by $(r(T) - r(T^*))/(n-1)$, which corresponds to a factor not larger than $1 - 1/n$. This holds independently of previous good edge-exchanges. After ρ good edge-exchanges, the expected value of the difference $r(T) - r(T^*)$ is at most $(1 - 1/n)^\rho \cdot (r(T) - r(T^*))$.

Since $r(T) \leq (n-1) \cdot m$, where m is the largest edge rank, and $r(T^*) \geq 1$, we obtain the upper bound $(1 - 1/n)^\rho \cdot n \cdot m$. If $\rho = \lceil (\ln 2) \cdot n \cdot (\log(n \cdot m) + 1) \rceil$, this bound is at most 1/2. By Markov's inequality, the probability that the bound is less than 1 is at least 1/2. Since $r(T) - r(T^*)$ is an integer, the probability of having found the MST is at least 1/2. Repeating these arguments, the expected number of good steps until the MST is found is bounded by $2\rho = O(n \log(n \cdot m)) = O(n \log n)$. ■

So far, we have assumed that each good edge-exchange is equally likely. This does not hold under biased edge-selection. If good edge-exchanges often involve the inclusion of edges whose selection probability is less than the uniform selection probability $1/m$, the required number of good edge-exchanges may be considerably larger. Since we consider only non-degenerate MSTs, whose edges have selection probabilities $q_A(r)$ of at least $1/m$, and since the proof does not rely on the specifically chosen dummy edge-exchanges (every edge-exchange that does not change the current tree can be considered a dummy edge-exchange), the assumption that a good edge-exchange decreases $r(T) - r(T^*)$ by a factor not larger than $1 - 1/n$ is still valid.

To obtain an upper bound for the expected optimization time, we multiply the required number of good edge-exchanges by the expected waiting time for each such exchange.

Recall that the set of edges to be inserted by all $n-1$ good edge-exchanges is identical to T^*, and let $p_{\mathrm{I}}(e^*)$ be the average probability of selecting a specific edge $e^* \in T^*$ for insertion. When choosing an edge uniformly at random from $E - T$, as in NA, UU, and UG,

$$p_{\mathrm{I}}^{\mathrm{NA}}(e^*) = p_{\mathrm{I}}^{\mathrm{UU}}(e^*) = p_{\mathrm{I}}^{\mathrm{UG}}(e^*) = \\ = \frac{1}{m - n + 1} = \Theta(1/n^2). \qquad (26)$$

When choosing the edge to be inserted according to the approximately optimal edge-selection strategy, as in BU and BG,

$$p_{\mathrm{I}}^{\mathrm{BU}}(e^*) = p_{\mathrm{I}}^{\mathrm{BG}}(e^*) \approx \frac{1}{EX^*(e^*)} = \Theta(1/n) \qquad (27)$$

according to corollary 1.

Assume an edge $e^* \in T^*$ has been selected for insertion into T, and an edge e' is to be selected for removal from T. If e^* is already contained in T, no matter which edge will be chosen for e', the tree T will remain unchanged: Either $e' = e^*$ and a (good) dummy edge-exchange is performed, or the resulting edge set is not a spanning tree and therefore not accepted by the (1+1)-EA. In case of the NA-variant, the probability of performing the dummy edge-exchange is

$$p_{\mathrm{R}}^{\mathrm{NA}}(e^*) = 1/(n-1) = \Theta(1/n). \qquad (28)$$

In the other variants, the edge to be removed is always $e' = e^*$, so that

$$p_{\mathrm{R}}^{\mathrm{UU}}(e^*) = p_{\mathrm{R}}^{\mathrm{UG}}(e^*) = p_{\mathrm{R}}^{\mathrm{BU}}(e^*) = p_{\mathrm{R}}^{\mathrm{BG}}(e^*) = 1. \qquad (29)$$

If $e^* = (u, v)$ is not yet in T, there exists exactly one edge $e'' \notin T^*$ on the path connecting u and v whose removal results in a good edge-exchange. The probability of choosing this edge as e' depends on the edge-exchange strategy. When choosing an edge uniformly from T as in the NA-case,

$$p_{\mathrm{R}}^{\mathrm{NA}}(e'') = 1/(n-1) = \Theta(1/n). \qquad (30)$$

When choosing an edge uniformly from the path connecting u and v, as in UU and BU, the probability depends on the length of the path, which is $n - 1 = O(n)$ in the worst case. However, if we assume that T is non-degenerate, as it is with overwhelming probability in the average case, the diameter and the expected length of the path are $O(\sqrt{n})$. Thus,

$$p_{\mathrm{R}}^{\mathrm{UU}}(e'') = p_{\mathrm{R}}^{\mathrm{BU}}(e'') = \Omega(1/\sqrt{n}). \qquad (31)$$

When the edge to be removed is greedily selected as in UG and BG, the maximum possible weight decrease is achieved; the (1+1)-EA either performs a good edge-exchange or an edge-exchange resulting in an even larger weight decrease, which we may consider an alternate good edge-exchange. Therefore,

$$p_{\mathrm{R}}^{\mathrm{UG}}(e'') = p_{\mathrm{R}}^{\mathrm{BG}}(e'') = 1. \qquad (32)$$

Note that $p_{\mathrm{R}}(e'') \leq p_{\mathrm{R}}(e^*)$ for all mutation variants. These observations lead to the following result.

Theorem 3: Excluding preprocessing, the total expected time for the (1+1)-EA to find the MST for average case instances is bounded above depending on its mutation operator:

$$EX(t_{\mathrm{MST}}^{\mathrm{NA}}) = O(n^3 \log n) \tag{33}$$
$$EX(t_{\mathrm{MST}}^{\mathrm{UU}}) = O(n^3 \log n) \tag{34}$$
$$EX(t_{\mathrm{MST}}^{\mathrm{UG}}) = O(n^{5/2} \log n) \tag{35}$$
$$EX(t_{\mathrm{MST}}^{\mathrm{BU}}) = O(n^2 \log n) \tag{36}$$
$$EX(t_{\mathrm{MST}}^{\mathrm{BG}}) = O(n^{3/2} \log n) \tag{37}$$

Proof: Since there are always $n-1$ good mutations, the probability of performing a good mutation (or an even better one) is bounded below by $(n-1) \cdot p_{\mathrm{I}}(e^*) \cdot p_{\mathrm{R}}(e'')$. This yields the expected waiting time

$$EX(t_1) = O\left(\frac{t_{\mathrm{mut}}}{n \cdot p_{\mathrm{I}}(e^*) \cdot p_{\mathrm{R}}(e'')}\right). \tag{38}$$

for performing a good mutation. Therefore, an upper bound for the total expected waiting time to find the MST when starting from a random spanning tree is

$$EX(t_{\mathrm{MST}}) = O(EX(t_1) \cdot n \log n) = \\ = O\left(\frac{t_{\mathrm{mut}} \cdot \log n}{p_{\mathrm{I}}(e^*) \cdot p_{\mathrm{R}}(e'')}\right). \tag{39}$$

Replacing $p_{\mathrm{I}}(e^*)$, $p_{\mathrm{R}}(e'')$, and t_{mut} by the specific complexities above yields the five results. ■

With BU and BG, the (1+1)-EA must begin by sorting the graph's edges and determining their ranks. This takes time that is $O(n^2 \log n)$ and dominates the EA's running time for these two mutation variants. Note that a classical implementation of Kruskal's algorithm in which all edges are initially sorted also has time $O(n^2 \log n)$, and there exist randomized algorithms that identify a MST in expected times that are $O(n^2)$ [20].

V. Selection Strategies for Edge-Insertion in Mutation

Assume an evolutionary algorithm that seeks a constrained subgraph of minimum total weight in a complete graph G, as in the problems listed in the introduction. The EA's mutation operator inserts a new edge into a feasible solution and guarantees the offspring's feasibility by applying a problem-dependent repair such as the removal of another edge in case of the MSTP. In the experiments the next sections describe, we compare the following strategies for selecting a new edge to be inserted.

Uniform Edge-Selection (UNIF)

This method corresponds to the edge-selection strategy used in the UU and UG mutation variants of Sect. IV. The new edge is randomly chosen with probability $q_U(r) = 1/m$ from E.

Approximately Optimal Edge-Selection (OPTEX)

The new edge is selected according to the approximately optimal selection probabilities $q_A(r)$, as in the BU and BG mutation strategies of Sect. IV.

To perform this edge-selection efficiently in practice, we derive a random edge-rank $\mathcal{R} \in \{1, 2, \ldots, m\}$ from a uniformly distributed random number $\mathcal{U} \in [0, 1)$.

In order to ensure that R has the approximate probability density $q_A(r)$ of Theorem 2, we use the corresponding cumulative distribution function $F(r)$:

$$F(r) = \sum_{i=1}^{r} q_A(i) \approx \sum_{i=1}^{r} a^{\frac{i}{2}} \left(\frac{1}{\sqrt{a}} - 1\right) \\ = \frac{\sqrt{a} - a^{(r+1)/2}}{1 - \sqrt{a}} \left(\frac{1}{\sqrt{a}} - 1\right) \\ = 1 - a^{r/2} = 1 - \left(\frac{|S|}{|S|+1}\right)^{\frac{r}{2}}. \tag{40}$$

The inverse of $F(r)$ is

$$r = \frac{2 \log(1 - F(r))}{\log |S| - \log(|S|+1)}. \tag{41}$$

$\mathcal{R}$ can be calculated from $\mathcal{U}$ by setting $F(r) = \mathcal{U}$ in (41) and rounding:

$$\mathcal{R} = \left\lfloor \frac{2 \log(1 - \mathcal{U})}{\log |S| - \log(|S|+1)} \right\rfloor \bmod m + 1. \tag{42}$$

Finding the modulus and adding one ensures that $\mathcal{R}$ will be a valid edge rank.

Proportional Edge-Selection (PROPP)

Each edge is selected with probability $q_P(r) = p(r)/|S| \approx a^r/|S|$. This operator's implementation uses a uniform random number $\mathcal{U}$ transformed by the inverse of the distribution function:

$$F(r) = \sum_{i=1}^{r} \frac{a^i}{|S|} = 1 - \left(\frac{|S|}{|S|+1}\right)^r. \tag{43}$$

This yields

$$\mathcal{R} = \left\lfloor \frac{\log(1 - \mathcal{U})}{\log |S| - \log(|S|+1)} \right\rfloor \bmod m + 1. \tag{44}$$

Normal-Distribution-Based Edge-Selection (N_β)

This edge-selection strategy is based on normal distributions as proposed in [21]. The rank of a selected edge is

$$\mathcal{R} = \lfloor |\mathcal{N} \cdot \beta \cdot n| \rfloor \bmod m + 1, \tag{45}$$

where $\mathcal{N}$ is a normally distributed random number with mean zero and standard deviation one. β controls the bias towards low-cost edges.

Inverse-Weight-Proportional Edge-Selection (INVW)

The probability of each edge $e \in E$ is inversely proportional to its weight $w(e)$. Greffenstette used this technique for choosing edges during recombination in a genetic algorithm for the TSP [6]. This selection can be implemented efficiently by applying binary search to an array of cumulated weights $w_c(e_i) = \sum_{j=1}^{i} w(e_j)$, $i = 1, \ldots, m$.

VI. Empirical Results for the (1+1)-EA on the MSTP

To support the theoretical results from Sect. IV and to compare the approximately optimal edge-selection strategy to the others presented in the previous section, we performed the following experiments with the (1+1)-EA.

Fifty uniform instances and fifty Euclidean instances of the MSTP were randomly created on $n = 20$, 50, 100, 200, 500, and $1\,000$ vertices. We applied the (1+1)-EA to all these instances, using each strategy for selecting a new edge with both uniform and greedy selection of the edge to be removed. For the normal-distribution-based edge-selection N_β, β was set to 0.75, 1, 1.5, 2, and 3. A run was terminated when either the MST was found or 10^7 iterations had been performed.

Table III lists the median numbers of iterations until termination. In addition, Table IV shows the corresponding error probabilities in percent for the hypothesis that on average OPTEX yields a smaller number of iterations than each of the other edge-selection strategies. These error probabilities were determined by paired one-sided Wilcoxon rank-sum tests.

For all mutation variants, greedy edge removal allows the EA to use substantially fewer iterations. When using OPTEX, it was always able to find the MST in fewer than 10^7 iterations; the increase in the number of iterations with respect to the problem size is only moderate. This could be expected; the derived upper bounds for the expected number of iterations are $O(n^2 \log n)$ and $O(n^{3/2} \log n)$ for uniform random and greedy edge removal, respectively.

As the small error probabilities in Table IV indicate, OPTEX speeds up the EA in comparison to the other edge-selection strategies, in most cases with high statistical significance. It always outperforms UNIF, N_3, and INVW with error probabilities less than 0.1%. On the smaller instances, PROPP, $\mathrm{N}_{0.75}$, N_1, and $\mathrm{N}_{1.5}$ were occasionally better than OPTEX, in particular on uniform instances. This sometimes poorer performance of OPTEX on uniform instances can be explained by the error resulting from the approximation $p_A(r)$ of the probabilities $p_E(r)$ with which an edge of a certain rank appears in the optimum solution. On uniform instances, $p_A(r)$ tends to underestimate $p_E(r)$ for $r < n$ and overestimate $p_E(r)$ for $r \geq n$; note Figs. 1 and 2.

Generally, OPTEX is almost always superior on Euclidean instances and on large uniform instances.

VII. Experiments on the 3-MSTP

We now evaluate biased mutation in more sophisticated EAs for two NP-hard problems, the d-MSTP and the TSP. We consider the d-MSTP first, with the maximum degree d set to three.

The EA is the steady-state algorithm described in [21], and it represents candidate spanning trees as sets of edges. Random spanning tree generation based on Kruskal's algorithm fills the population with feasible initial solutions [22]. The EA selects parents for crossover in binary tournaments with replacement, and crossover builds an offspring degree-constrained spanning tree from the union of the parents' edge-sets. Only if necessary, edges not appearing in the parents are added in order to satisfy the degree-constraint. Each (feasible) offspring is always generated by applying crossover to two parents, then mutating the resulting tree.

Mutation selects an edge to be included in a tree according to one of Sect. V's five strategies. To guarantee that offspring satisfy the degree constraint, the following two special cases extend the simple edge-exchange mutation from the unconstrained MSTP.

- If the insertion of the selected edge violates the degree constraint at both of its end-vertices, the edge is not suitable for insertion and is therefore discarded; another edge is selected.
- If the new edge violates the degree constraint at one of its end-vertices, the edge to be removed must be the other edge incident to this vertex in the introduced cycle.

In all other cases, the edge to be removed is always chosen according to the U-variant; i.e., at random from all edges on the induced cycle, excluding the newly inserted edge. The greedy variant, which always removes the edge of highest weight, is not useful here; in preliminary experiments, this approach almost always led to premature convergence at poor, locally optimum solutions. On the unconstrained MSTP, greedy selection of the edge to be removed exploits the fact that a targeted edge could not belong to an optimum solution; this condition does not hold in the d-MSTP.

Each offspring replaces the worst solution in the population except when it is identical to an existing solution; duplicates are discarded to maintain diversity.

We considered 50 random Euclidean instances of sizes $n = 50$, 100, and 200. For all these instances, we determined optimum solutions by branch-and-cut, again using ABACUS [13] and CPLEX 8.1. The EA's population size was $2\,n$, and the EA terminated if an optimum solution had been reached or the number of evaluations exceeded $5\,000\,n$.

We performed 50 runs on each instance with each mutation variant. Table V shows, for each size n and each operator, the percentage of runs that identified optimum solutions and the average number of evaluations over all runs.

In all but the uniform case with $n = 200$, OPTEX solved the largest number of instances to optimality. In that one case, PROPP, $\mathrm{N}_{0.75}$, and N_1 solved one more instance. With respect to average numbers of required iterations, OPTEX again outperformed the other mutation variants in every case but one. For uniform instances of size $n = 50$, N_1 gave better results. Table VI shows error probabilities for the hypotheses that on average, OPTEX yields smaller numbers of iterations than each other mutation method; again, these values were calculated by paired one-sided Wilcoxon rank-sum tests. In most cases, these error probabilities are less than 0.1%, indicating high significance.

VIII. Experiments on the TSP

The EA framework used for the d-MSTP was also applied to the TSP. The EA represented candidate tours as permutations of the vertices. It applied unbiased random initialization, standard edge recombination crossover (ERX) [23], and mutation biased according to the five edge-selection strategies.

TABLE III

MEDIAN NUMBERS OF ITERATIONS REQUIRED TO FIND THE MST BY THE (1+1)-EA.

Mutation	Uniform random choice of edge to be removed (U)						Greedy choice of edge to be removed (G)					
	$n=20$	$n=50$	$n=100$	$n=200$	$n=500$	$n=1\,000$	$n=20$	$n=50$	$n=100$	$n=200$	$n=500$	$n=1\,000$
Uniform random instances												
UNIF	2 008	24 513	186 378	1 055 184	$>10^7$	$>10^7$	544	5 077	24 566	112 391	815 387	3 535 240
OPTEX	359	2 022	**6 561**	**24 344**	**119 389**	**396 044**	90	322	853	2 083	6 340	**14 360**
PROPP	290	**1 571**	8 099	25 789	164 687	499 716	60	268	740	2 029	7 764	18 124
$N_{0.75}$	198	2 921	48 129	510 473	$>10^7$	$>10^7$	**47**	433	4 221	33 599	1 011 781	3 413 491
N_1	**195**	1 631	9 681	86 262	755 918	$>10^7$	51	**234**	885	4 797	48 964	370 644
$N_{1.5}$	321	1 831	6 881	27 355	164 353	575 404	78	290	**722**	**1 909**	6 778	19 572
N_2	525	2 339	7 552	24 988	129 466	445 749	113	370	950	2 128	**6 087**	16 950
N_3	598	3 570	12 272	37 362	183 666	564 766	187	633	1 503	3 496	10 021	20 738
INVW	581	4 412	18 345	86 950	446 253	$>10^7$	111	684	2 132	6 238	22 821	59 729
Euclidean instances												
UNIF	1 324	20 294	106 648	845 683	9 772 113	$>10^7$	536	4 676	23 945	107 050	766 805	3 800 496
OPTEX	**292**	**1 826**	**7 933**	**23 342**	**175 904**	**643 915**	110	**422**	**1 094**	**2 836**	**8 774**	**19 924**
PROPP	301	5 420	25 388	108 910	666 924	2 934 556	**106**	557	1 667	6 122	24 578	50 717
$N_{0.75}$	2 159	985 295	$>10^7$	$>10^7$	$>10^7$	$>10^7$	432	90 346	2 453 202	$>10^7$	$>10^7$	$>10^7$
N_1	545	33 516	413 170	2 894 652	$>10^7$	$>10^7$	154	3 509	33 062	110 399	1 524 522	$>10^7$
$N_{1.5}$	373	4 988	24 649	161 841	1 411 807	$>10^7$	**106**	599	1 859	8 439	28 952	112 709
N_2	401	2 882	16 379	64 448	336 576	1 707 227	121	452	1 158	2 997	9 763	35 700
N_3	541	2 981	11 663	37 254	227 970	844 947	206	624	1 649	3 759	10 558	25 222
INVW	790	6 350	39 911	181 224	1 785 180	$>10^7$	209	1 455	5 484	18 841	93 982	286 671

TABLE IV

THE (1+1)-EA FOR THE MSTP: ERROR PROBABILITIES p_{err} [%] OF THE HYPOTHESIS THAT ON AVERAGE OPTEX YIELDS A SMALLER NUMBER OF ITERATIONS THAN THE OTHER MUTATION METHODS.

Mutation	Uniform random choice of edge to be removed (U)						Greedy choice of edge to be removed (G)					
	$n=20$	$n=50$	$n=100$	$n=200$	$n=500$	$n=1\,000$	$n=20$	$n=50$	$n=100$	$n=200$	$n=500$	$n=1\,000$
Uniform random instances												
UNIF	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PROPP	82.3	97.2	0.2	26.0	0.0	0.3	100.0	99.7	98.0	38.3	0.0	0.0
$N_{0.75}$	100.0	0.1	0.0	0.0	0.0	0.0	100.0	5.5	0.0	0.0	0.0	0.0
N_1	100.0	91.5	0.1	0.0	0.0	0.0	100.0	99.7	32.3	0.0	0.0	0.0
$N_{1.5}$	63.7	78.3	15.1	5.0	0.0	0.0	97.6	88.8	99.7	67.8	4.8	0.0
N_2	0.0	0.2	0.1	5.2	15.6	4.5	0.0	0.0	0.5	22.7	33.2	0.0
N_3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
INVW	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Euclidean instances												
UNIF	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PROPP	8.3	0.0	0.0	0.0	0.0	0.0	52.1	0.0	0.0	0.0	0.0	0.0
$N_{0.75}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N_1	0.0	0.0	0.0	0.0	0.0	0.0	2.3	0.0	0.0	0.0	0.0	0.0
$N_{1.5}$	0.3	0.0	0.0	0.0	0.0	0.0	80.2	0.0	1.0	0.0	0.0	0.0
N_2	0.0	0.0	0.0	0.0	0.0	0.0	8.3	0.8	2.8	13.7	6.3	0.0
N_3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
INVW	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

In comparison to the d-MSTP, incorporating specific edge-selection techniques into mutation in an EA for the TSP is more complex. A commonly used mutation operator for permutations is inversion. This operator can be modified to include a specific new edge selected by one of the above strategies: Invert the substring beginning after the selected edge's first vertex and ending with the selected edge's second vertex. For example, let $T = (a, b, c, d, e, f, g, h)$ be the tour to be mutated, and let (c, f) be the edge selected for insertion. The mutated tour is (a, b, c, f, e, d, g, h).

Note, however, that mutation for the TSP necessarily includes a second edge in addition to the selected one (in the above example, the edge (d, g)). This second edge cannot be chosen according to the edge-selection strategy, but depends on the first edge and the current tour. Furthermore, two edges that directly depend on the pair of inserted edges are removed ((c, d) and (f, g) in the example). These side-effects strongly influence the performance of biased mutation.

The experiments were performed on 50 random Euclidean instances of each size $n = 20$, 40, and 60, using a population of $2\,n$ tours. The EA halted when it reached an optimum solution or performed $5\,000\,n$ evaluations.

Tables VII and VIII list the results of these trials. In contrast to the d-MSTP, they do not in general show significant differ-

TABLE V
RESULTS OF THE EA FOR THE 3-MSTP WITH EACH MUTATION: PERCENTAGE OF RUNS THAT FOUND OPTIMUM SOLUTIONS (%-$hits$) AND AVERAGE NUMBERS OF ITERATIONS ($iter$).

Mutation	$n = 50$		$n = 100$		$n = 200$	
	%-$hits$	$iter$	%-$hits$	$iter$	%-$hits$	$iter$
Uniform random instances						
UNIF	54	173 522	2	498 396	0	1 000 000
OPTEX	**96**	46 265	**64**	**281 021**	5	**963 850**
PROPP	88	53 203	46	325 680	**6**	971 200
$N_{0.75}$	78	69 740	48	350 353	**6**	974 948
N_1	**96**	**35 979**	60	284 833	**6**	975 521
$N_{1.5}$	88	53 781	40	342 982	4	976 387
N_2	86	60 941	32	398 460	4	971 030
N_3	76	83 164	32	390 116	0	1 000 000
INVW	76	83 127	26	411 267	0	1 000 000
Euclidean instances						
UNIF	98	57 900	66	360 199	8	990 999
OPTEX	**100**	**11 698**	**100**	**49 088**	**96**	**230 771**
PROPP	**100**	16 877	88	118 942	78	411 953
$N_{0.75}$	48	149 643	20	432 160	6	967 138
N_1	68	96 968	46	314 193	46	715 735
$N_{1.5}$	94	25 005	82	166 909	64	503 273
N_2	**100**	15 007	90	92 152	76	404 871
N_3	98	16 216	94	87 970	66	468 987
INVW	98	24 980	86	169 252	46	824 540

TABLE VI
THE EA FOR THE 3-MSTP: ERROR PROBABILITIES p_{err} [%] OF THE HYPOTHESIS THAT ON AVERAGE OPTEX YIELDS SMALLER NUMBERS OF ITERATIONS THAN THE OTHER MUTATION METHODS.

Mutation	Uniform random instances			Euclidean instances		
	$n = 50$	$n = 100$	$n = 200$	$n = 50$	$n = 100$	$n = 200$
UNIF	0.0	0.0	0.0	0.0	0.0	0.0
PROPP	6.1	0.0	45.9	8.5	8.3	0.1
$N_{0.75}$	7.0	0.0	39.4	0.0	0.0	0.0
N_1	89.8	13.8	36.1	0.0	0.0	0.0
$N_{1.5}$	4.9	0.0	31.3	0.0	0.0	0.2
N_2	0.1	0.0	28.9	17.2	7.3	1.6
N_3	0.0	0.0	0.0	2.4	15.3	0.1
INVW	0.0	0.0	0.0	0.0	0.0	0.0

TABLE VII
RESULTS OF THE EA FOR THE TSP WITH EACH MUTATION ON EUCLIDEAN INSTANCES: PERCENTAGE OF RUNS THAT FOUND OPTIMUM SOLUTIONS (%-$hits$) AND AVERAGE NUMBERS OF ITERATIONS ($iter$).

Mutation	$n = 20$		$n = 40$		$n = 60$	
	%-$hits$	$iter$	%-$hits$	$iter$	%-$hits$	$iter$
UNIF	86	18 349	30	150 169	4	298 993
OPTEX	88	14 955	30	145 812	**10**	**276 288**
PROPP	**92**	**11 213**	18	165 133	8	278 601
$N_{0.75}$	84	19 805	12	176 786	4	294 130
N_1	88	14 756	14	172 602	4	288 515
$N_{1.5}$	90	14 069	30	144 882	6	282 637
N_2	90	12 720	**36**	**130 448**	4	295 192
N_3	88	13 899	22	159 152	6	283 035
INVW	86	16 958	22	159 466	4	289 819

TABLE VIII
THE EA FOR THE TSP: ERROR PROBABILITIES p_{err} [%] OF THE HYPOTHESIS THAT ON AVERAGE OPTEX NEEDS A SMALLER NUMBER OF ITERATIONS THAN THE OTHER MUTATION METHODS.

Mutation	$n = 20$	$n = 40$	$n = 60$
UNIF	0.0	30.5	10.8
PROPP	93.5	15.1	38.8
$N_{0.75}$	93.8	2.4	11.1
N_1	98.1	5.9	13.3
$N_{1.5}$	97.3	68.6	26.0
N_2	67.9	87.3	11.1
N_3	10.8	19.1	24.6
INVW	13.7	19.1	11.7

ences among the edge-selection methods. The only exception with high statistical significance is that OPTEX performs better than unbiased mutation when $n = 20$.

IX. CONCLUSION

We empirically analyzed the rank-based probabilities $p_E(r)$ with which edges appear in optimum solutions of Euclidean and uniform random instances of the MSTP of sizes 20 to 1 000, of the 3-MSTP and the 5-MSTP of sizes 20 to 100, and of the TSP of sizes 20 to 100. These probabilities are closely approximated by exponential functions $p_A(r)$ with absolute mean square errors less than 1% and relative mean square errors less than 10%.

Based on this approximation, we derived probabilities $q_A(r)$ for selecting edges to be incorporated into candidate solutions of an EA during mutation such that the average expected number of edge-selections until choosing an edge of an optimum solution is minimized. Using these probabilities, the expected number of edge-selections until each edge in an optimum solution is chosen is asymptotically linear in the number of vertices instead of $\Theta(n^2)$ with uniform selection.

Next, we analyzed a (1+1)-EA, with five variants of edge-exchange mutation, for the unconstrained MSTP. The edge to be inserted was selected either at random or according to the probabilities $q_A(r)$. The edge to be removed is chosen uniformly or greedily. We obtained upper bounds for expected running times of the (1+1)-EA on non-degenerate instances. When using the approximately optimal edge-selection strategy for choosing the edge to be inserted, the expected running time is dominated by the initial sorting of the edges to determine ranks, and the (1+1)-EA is, on a non-degenerate instance, asymptotically as fast, as a classical implementation of Kruskal's MST algorithm.

The edge-selection strategies were further compared to proportional, normal-distribution-based, and inverse-weight-proportional schemes on uniform and Euclidean MSTP instances with up to 1 000 vertices. The results support the theory and indicate that the new approximately optimal edge-selection scheme is in practice superior to the other methods.

Although we considered Euclidean and uniform random instances only, we conjecture that the same biasing towards low-weight edges also works well on other instance classes structured in different ways. A preliminary study on the "hard and misleading" instances from [1] supports this.

We further considered EAs with larger populations and recombination operators without specific biases for two NP-hard graph problems, the 3-MSTP and the TSP. With approximately optimal edge-selection probabilities, the EA for the 3-MSTP identified optimum solutions significantly more often and with fewer iterations. On the TSP, however, mutation that introduces one new edge introduces a second edge as well. While the first edge may be chosen according to specific probabilities, the second edge depends on the first edge and on the current tour, and two strongly dependent edges are removed. These side-effects overwhelm the differences between the various edge-selection strategies.

More generally, we conclude that for various problems, a detailed study of probabilities with which features appear in optimum or nearly optimum solutions may allow the derivation of theoretically well-justified biasing schemes. Mutation operators that are biased in this way are likely to find (near-)optimum solutions more often and more quickly. While we focused here on certain subset-selection problems, the basic idea is generally applicable. Future work will consider biased recombination operators and other application areas.

ACKNOWLEDGMENT

The authors thank the Austrian Science Fund (FWF) for supporting this work under grant P16263-N04 and the anonymous reviewers for their perceptive and helpful observations.

REFERENCES

[1] J. Knowles and D. Corne, “A new evolutionary approach to the degree constrained minimum spanning tree problem,” *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 2, pp. 125–134, 2000.

[2] M. Krishnamoorthy, A. T. Ernst, and Y. M. Sharaiha, “Comparison of algorithms for the degree constrained minimum spanning tree,” *Journal of Heuristics*, vol. 7, pp. 587–611, 2001.

[3] N. Deo and P. Micikevicius, “A heuristic for a leaf constrained minimum spanning tree problem,” *Congressus Numerantium*, vol. 141, pp. 61–72, 1999.

[4] B. A. Julstrom, “Codings and operators in two genetic algorithms for the leaf-constrained minimum spanning tree problem,” *International Journal of Applied Mathematics and Computer Science*, vol. 14, no. 3, pp. 385–396, 2004.

[5] I. Ljubić and G. R. Raidl, “A memetic algorithm for minimum-cost vertex-biconnectivity augmentation of graphs,” *Journal of Heuristics*, no. 9, pp. 401–427, 2003.

[6] J. J. Grefenstette, “Incorporating problem specific knowledge into genetic algorithms,” in *Genetic Algorithms and Simulated Annealing*, L. Davis, Ed. Morgan Kaufmann, 1987, pp. 42–60.

[7] B. A. Julstrom, “Very greedy crossover in a genetic algorithm for the Traveling Salesman Problem,” in *Proceedings of the 1995 ACM Symposium on Applied Computing*, K. M. George *et al.*, Eds. ACM Press, 1995, pp. 324–328.

[8] G. R. Raidl and B. A. Julstrom, “Edge-sets: An effective evolutionary coding of spanning trees,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, 2003.

[9] F. Rothlauf and C. Tzschoppe, “On the bias and performance of the edge-set encoding,” Department of Business Administration and Information Systems, University of Mannheim, Germany, Tech. Rep. 11/2004, 2004.

[10] B. A. Julstrom and G. R. Raidl, “Weight-biased edge-crossover in evolutionary algorithms for two graph problems,” in *Proceedings of the 16th ACM Symposium on Applied Computing*, G. Lamont *et al.*, Eds. ACM Press, 2001, pp. 321–326.

[11] G. R. Raidl, G. Kodydek, and B. Julstrom, “On weight-biased mutation for graph problems,” in *Parallel Problem Solving from Nature – PPSN VII*, ser. LNCS, J. J. M. Guervos *et al.*, Eds., vol. 2439. Springer, 2002, pp. 204–213.

[12] J. B. Kruskal, “On the shortest spanning subtree of a graph and the traveling salesman problem,” *Proceedings of the American Mathematics Society*, vol. 7, no. 1, pp. 48–50, 1956.

[13] S. Thienel, “ABACUS – A Branch-And-CUt System,” Ph.D. dissertation, University of Cologne, Cologne, Germany, 1995.

[14] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge University Press, 1995.

[15] F. Neumann and I. Wegener, “Randomized local search, evolutionary algorithms, and the minimum spanning tree problem,” in *Genetic and Evolutionary Computation – GECCO 2004*, ser. LNCS, K. Deb *et al.*, Eds., vol. 3102. Springer, 2004, pp. 713–724.

[16] A. Rényi and G. Szekeres, “On the height of trees,” *Journal of the Australian Mathematical Society*, vol. 7, pp. 497–507, 1967.

[17] G. Szekeres, “Distribution of labelled trees by diameter,” in *Lecture Notes in Mathematics*, 1983, vol. 1036, pp. 392–397.

[18] A. Broder, “Generating random spanning trees,” in *IEEE 30th Annual Symposium on Foundations of Computer Science*. IEEE Press, 1989, pp. 442–447.

[19] E. W. Mayr and C. G. Plaxton, “On the spanning trees of weighted graphs,” *Combinatorica*, vol. 12, pp. 433–447, 1992.

[20] D. R. Karger and R. E. Tarjan, “A randomized linear time algorithm for finding minimum spanning trees,” *Journal of the ACM*, vol. 42, no. 2, pp. 321–328, 1995.

[21] G. R. Raidl, “An efficient evolutionary algorithm for the degree-constrained minimum spanning tree problem,” in *Proceedings of the 2000 IEEE Congress on Evolutionary Computation*, C. Fonseca, J.-H. Kim, and A. Smith, Eds. IEEE Press, 2000, pp. 104–111.

[22] B. A. Julstrom and G. R. Raidl, “Initialization is robust in evolutionary algorithms that encode spanning trees as sets of edges,” in *Proceedings of the 2002 ACM Symposium on Applied Computing*, G. Lamont, J. Llorens, J. Carrol, *et al.*, Eds. ACM Press, 2002, pp. 547–552.

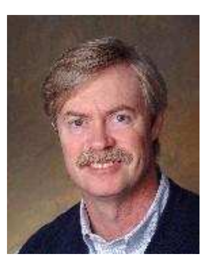
[23] D. Whitley, T. Starkweather, and D. Fuquay, “Scheduling problems and traveling salesman: The genetic edge recombination operator,” in *Proceedings of the Third International Conference on Genetic Algorithms*, J. D. Schaffer, Ed. Morgan Kaufmann, 1989, pp. 133–140.

Günther R. Raidl received the M.S. and Ph.D. degrees in computer science from the Vienna University of Technology, Vienna, Austria, in 1992 and 1994, respectively, and habilitated in 2003.

He is currently Professor in the Institute of Computer Graphics and Algorithms at the Vienna University of Technology and heading the Algorithms and Data Structures Group. His main research interests lie in the area of combinatorial optimization with metaheuristics, such as evolutionary algorithms, and exact approaches, including integer linear programming techniques. He has authored or coauthored more than 50 papers and coedited more than 10 books.

Gabriele Koller received the M.S. and Ph.D. degrees in business informatics from the Vienna University of Technology and University of Vienna, Austria, in 1996 and 2001, respectively.

She is currently university assistant and lecturer in the Institute of Computer Graphics and Algorithms at the Vienna University of Technology. Her main research interests lie in heuristic optimization techniques, bioinformatics, and algorithms for German word hyphenation. She has so far authored or coauthored 12 refereed papers.



Bryant A. Julstrom received the M.S. degree in mathematics and the M.S. and Ph.D. degrees in computer science from The University of Iowa, Iowa City, Iowa, in 1973, 1982, and 1987, respectively.

He is currently Professor in the Department of Computer Science at St. Cloud State University, St. Cloud, Minnesota. His research addresses codings and operators in evolutionary algorithms for problems of combinatorial optimization. He has authored or coauthored more than 50 refereed papers.