On Weight-Biased Mutation for Graph Problems

Günther R. Raidl[1], Gabriele Kodydek[1], and Bryant A. Julstrom[2]

[1] Vienna University of Technology, Vienna, Austria
[2] St. Cloud State University, St. Cloud, MN, U.S.A.
{raidl|kodydek}@ads.tuwien.ac.at, julstrom@eeyore.stcloudstate.edu

Abstract. Many graph problems seek subgraphs of minimum weight that satisfy the problems' constraints. Examples include the degree-constrained minimum spanning tree and traveling salesman problems. Low-weight edges predominate in optimal solutions to these problems, and the performance of evolutionary algorithms for them is often improved by biasing their operators to favor these edges. From the distributions of edges' ranks in optimal solutions to these two problems, we identify probabilities for edges that minimize the average expected time until mutation chooses them for inclusion in a solution. On instances of the degree-constrained minimum spanning tree problem, an evolutionary algorithm performs better with this operator than with alternative mutations. These results are not replicated on instances of the traveling salesman problem, where the inclusion of one edge in a tour requires the inclusion of another dependant edge.

1 Introduction

Given a weighted, undirected graph, many graph problems seek a subset S of the graph's edges that satisfies a set of constraints and has minimum total weight. These include the familiar traveling salesman problem (TSP), in which the edges in S form a Hamiltonian tour; the degree-constrained minimum spanning tree problem (d-MSTP) [6, 7], in which S is a spanning tree with degree no greater than a bound d; the leaf-constrained spanning tree problem [1], in which S is a spanning tree with at least L leaves; the biconnectivity augmentation problem [8], in which S augments a spanning tree so that the resulting network is 2-connected; and many others.

Some of these problems, such as the unconstrained minimum spanning tree problem and the identification of the shortest path between two vertices, can be solved to optimality in polynomial time. Most, including those listed above, are NP-hard. It is not likely that fast algorithms exist to solve these problems exactly, so we turn to heuristics, including evolutionary algorithms (EAs).

It is not surprising—and we verify below—that low-weight edges predominate in optimal solutions to such problems. This suggests that in EAs, crossover and mutation, which build representations of novel solutions from existing representations, should be biased so as to favor edges of lower weight. Several researchers have investigated such schemes [3, 4, 9, 10].

This work is supported by the Austrian Science Fund (FWF), grant P13602–INF.

Among them, Julstrom and Raidl examined weight-biased crossover operators in EAs for the TSP and the d-MSTP [5]; favoring low-weight edges improved the performance of these algorithms. We extend that inquiry to mutation and derive probabilities for selecting edges to be incorporated into candidate solutions. These probabilities are optimal in the sense that they minimize the expected time to include edges of optimal solutions. For the d-MSTP, we compare a mutation based on this analysis to four others. This theoretically approximately optimal scheme increases the probability of finding optimal solutions and reduces the number of iterations usually used. Applied to the TSP, the advantages of weight-biased approaches are generally smaller because mutation that introduces one edge into a tour necessarily introduces a second as well.

2 Distribution of Edges in Optimal Solutions

It is reasonable that optimally low-weight trees, tours, and other structures in weighted graphs should contain high proportions of low-weight edges. This section confirms and quantifies this observation for the degree-constrained minimum spanning tree and traveling salesman problems on complete graphs $G = (V, E)$ with $n = |V|$ nodes and $m = |E| = n \cdot (n-1)/2$ edges. Let S be the set of edges in a solution, so that $|S| = n - 1$ for the d-MSTP and $|S| = n$ for the TSP.

We examine two kinds of instances of both problems. Euclidean instances consist of distinct points chosen randomly in a square region of the plane; edge weights are the Euclidean distances between each pair of points. Uniform instances consist of edge weights chosen randomly and independently from a specified interval. 1 000 instances of each type were generated with $n = 20$, 50, and 100 nodes. For the d-MSTP, the degree bound d was set to three for the Euclidean instances; note that for such instances there always exists an unconstrained minimum spanning tree whose degree does not exceed five. On the uniform instances, we consider $d = 3$ and $d = 5$.

All these instances were solved to optimality by an algorithm found in the ABACUS branch-and-cut solver [11]. We assign each edge a rank r, $1 \le r \le m$, by sorting the edges of an instance according to increasing weights (ties are broken randomly). Figure 1 plots the probabilities $p(r)$ that an edge of rank r appears in an optimal solution. Only the portions of the curves where $p(r)$ is significantly larger than zero are plotted.

Note that the sum of the probabilities $p(r)$ is $|S|$:

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

As predicted, the optimal solutions consist mostly of low-rank—*i.e.*, short—edges. Further, for each kind of problem and each fraction $k \in (0, (n-1)/2]$, the probability $p(\lceil k \cdot n \rceil)$ is approximately constant across all values of $n \gg 1$.

Table 1 illustrates this by listing, for each problem kind and size, the number R of least-cost edges among which $\alpha = 50$, 90, and 99 percent of the edges of an

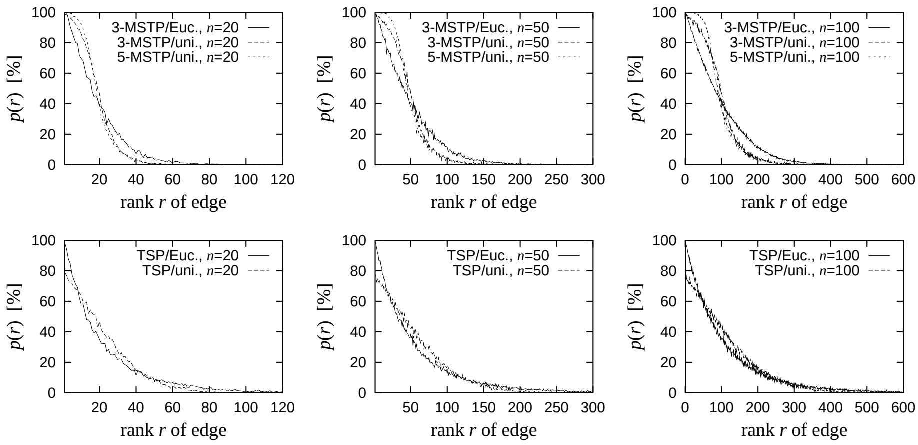

Fig. 1. The probability that an edge appears in an optimal solution as a function of its rank, shown for the 3-MSTP, the 5-MSTP, and the TSP on Euclidean and uniform instances of size $n = 20$, 50, and 100.

Table 1. Numbers R of least-cost edges for each problem class and size $n = 20, 50, 100$ among which $\alpha = 50\%$, 90%, and 99% of optimal solutions' edges are found and corresponding fractions $k = R/n$.

		3-MSTP/Euc.			3-MSTP/uni.			5-MSTP/uni.			TSP/Euc.			TSP/uni.		
$\alpha \setminus n$		20	50	100	20	50	100	20	50	100	20	50	100	20	50	100
50%	R	12	31	63	11	27	54	10	25	51	15	37	72	16	40	80
	k	0.6	0.6	0.6	0.6	0.5	0.5	0.5	0.5	0.5	0.8	0.7	0.7	0.8	0.8	0.8
90%	R	33	89	179	24	63	126	23	60	120	53	134	257	41	107	217
	k	1.7	1.8	1.8	1.2	1.3	1.3	1.2	1.2	1.2	2.7	2.7	2.6	2.1	2.1	2.2
99%	R	63	165	323	41	113	228	40	110	223	107	296	586	67	183	373
	k	3.2	3.3	3.2	2.1	2.3	2.3	2.0	2.2	2.2	5.4	5.9	5.9	3.4	3.7	3.7

optimal solution are located, i.e., the smallest rank R for which the cumulated probability

$$\sum_{r=1}^{R} p(r) \geq \alpha \cdot |S| \,. \tag{2}$$

Table 1 also shows corresponding fractions $k = R/n$, which are quite independent of n for each problem class and each α.

An effective heuristic mutation operator should introduce edges depending on the probabilities with which they appear in optimal solutions. To do this, we identify a closed-form expression $p_A(r)$ that approximates $p(r)$.

Fig. 1 shows that $p(r)$ decreases approximately exponentially with r, particularly in the Euclidean instances of the two problems. Thus we choose

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

The base a should be chosen so that

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

Since a^{m+1} is negligible for problems of even moderate size, we ignore it to obtain

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

Fig. 2(a) in Sect. 3.2 plots $p_A(r) = a^r$ with $a = |S|/(|S|+1)$ for the 3-MSTP instances with 100 nodes. It approximates the empirical probabilities $p(r)$ with high accuracy; the mean square error is less than 0.076%. For the 100-node Euclidean instances of the TSP, the mean square error is less than 0.014%. For uniform instances, the error is slightly larger.

3 Optimal Edge-Selection Probabilities

In genetic algorithms, mutation is understood to (re)introduce into the population novel or lost genetic material. In graph problems like the d-MSTP and the TSP, the $m = \binom{n}{2}$ edges of the graph comprise the pool from which this material is drawn.

Purely random mutation chooses each edge to include in a solution according to uniform probabilities; each edge may be chosen with probability $1/m$. We apply the analysis of Section 2 to identify non-uniform probabilities, associated with the edges' ranks, that are optimal in the following sense: Over all edges e^* in an optimal 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. The expected number of selections until e_r is chosen for the first time is

$$EX(e_r) = 1/q(r)\,. \tag{6}$$

Let e^* be an edge in an optimal solution S^*. The probability that e^* has rank r ($1 \le r \le m$) is $p(r)/|S|$, where $p(r)$ is the probability that e_r appears in an optimal solution. The expected number of edge selections until e^* is chosen for the first time is the weighted sum

$$EX(e^*) = \sum_{r=1}^{m} \frac{p(r)/|S|}{q(r)} = \frac{1}{|S|} \sum_{r=1}^{m} \frac{p(r)}{q(r)}\,. \tag{7}$$

Because $\sum_{r=1}^{m} q(r) = 1$, we can replace $q(m)$ by $1 - \sum_{i=1}^{m-1} q(i)$ in (7) and write

$$EX(e^*) = \frac{1}{|S|} \left(\sum_{r=1}^{m-1} \frac{p(r)}{q(r)} + \frac{p(m)}{1 - \sum_{i=1}^{m-1} q(i)} \right) . \tag{8}$$

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

$$\begin{aligned}
\frac{\partial EX(e^*)}{\partial q(1)} &= \frac{1}{|S|} \left(-\frac{p(1)}{q(1)^2} + \frac{p(m)}{(1 - \sum_{i=1}^{m-1} q(i))^2} \right) = 0 \\
\frac{\partial EX(e^*)}{\partial q(2)} &= \frac{1}{|S|} \left(-\frac{p(2)}{q(2)^2} + \frac{p(m)}{(1 - \sum_{i=1}^{m-1} q(i))^2} \right) = 0 \\
&\cdots \\
\frac{\partial EX(e^*)}{\partial q(m-1)} &= \frac{1}{|S|} \left(-\frac{p(m-1)}{q(m-1)^2} + \frac{p(m)}{(1 - \sum_{i=1}^{m-1} q(i))^2} \right) = 0
\end{aligned} \tag{9}$$

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

$$\frac{p(1)}{q(1)^2} = \frac{p(2)}{q(2)^2} = \cdots = \frac{p(m-1)}{q(m-1)^2} = \frac{p(m)}{(1 - \sum_{i=1}^{m-1} q(i))^2} = \frac{p(m)}{q(m)^2} . \tag{10}$$

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

$$q(r) = \sqrt{\frac{p(r)}{\varphi}} \tag{11}$$

and since

$$\sum_{i=1}^{m} q(i) = 1 = \frac{1}{\sqrt{\varphi}} \sum_{i=1}^{m} \sqrt{p(i)} , \tag{12}$$

we conclude that

$$\varphi = \left(\sum_{i=1}^{m} \sqrt{p(i)} \right)^2 \quad \text{and} \quad q(r) = \frac{\sqrt{p(r)}}{\sum_{i=1}^{m} \sqrt{p(i)}} . \tag{13}$$

3.1 $EX(e^*)$ for Three Edge-Selection Strategies

The optimal edge-selection probabilities $q(r)$ identified in (13), when substituted into equation (7), yield the following average expected number of edge-selections until an edge e^* of an optimal solution is chosen:

$$EX^*(e^*) = \frac{1}{|S|} \sum_{r=1}^{m} \frac{p(r)}{\sqrt{p(r)} \Big/ \sum_{i=1}^{m} \sqrt{p(i)}} = \frac{1}{|S|} \left(\sum_{r=1}^{m} \sqrt{p(r)} \right)^2 . \tag{14}$$

We replace $p(r)$ by the approximation $p_A(r) = a^r$ to obtain

$$EX^*(e^*) \approx \frac{1}{|S|}\left(\sum_{r=1}^{m}\sqrt{a^r}\right)^2 = \frac{1}{|S|}\left(\frac{\sqrt{a}-a^{(m+1)/2}}{1-\sqrt{a}}\right)^2 . \tag{15}$$

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

$$EX^*(e^*) \approx \frac{a}{|S|\,(1-\sqrt{a})^2} = \frac{1}{(|S|+1)\left(1-\sqrt{\frac{|S|}{|S|+1}}\right)^2} = \tag{16}$$

$$= \left(\sqrt{|S|}+\sqrt{|S|+1}\right)^2 . \tag{17}$$

Thus, $EX^*(e^*) = O(|S|) = O(n)$.

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 $\sum_{r=1}^{m} p(r) = |S|$,

$$EX^U(e^*) = \frac{1}{|S|}\sum_{r=1}^{m}\frac{p(r)}{1/m} = \frac{m}{|S|}\sum_{r=1}^{m} p(r) = m\,. \tag{18}$$

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

$$EX^P(e^*) = \frac{1}{|S|}\sum_{r=1}^{m}\frac{p(r)}{p(r)/|S|} = \frac{|S|}{|S|}\sum_{r=1}^{m} 1 = m\,. \tag{19}$$

For both, uniform and $p(r)$-proportional probabilities, $EX(e^*) = m = O(n^2)$, while for the optimal probabilities, $EX^*(e^*) = O(n)$.

3.2 Approximately optimal edge-selection probabilities

Replacing $p(r)$ by the approximation $p_A(r) = a^r$ in (13) yields a closed-form expression for the optimal edge-selection probabilities $q_A(r)$:

$$q_A(r) = \frac{\sqrt{p_A(r)}}{\sum_{i=1}^{m}\sqrt{p_A(i)}} = \frac{\sqrt{a^r}}{\sum_{i=1}^{m}\sqrt{a^i}} = \frac{\sqrt{a^r}}{\frac{\sqrt{a}-a^{(m+1)/2}}{1-\sqrt{a}}} = \frac{(1-\sqrt{a})\,a^{r/2}}{\sqrt{a}-a^{(m+1)/2}} \tag{20}$$

Again, $a^{(m+1)/2} \ll \sqrt{a}$, and we ignore it. Again, we replace a with $|S|/(|S|+1)$ according to (5). Thus:

$$q_A(r) \approx \frac{(1-\sqrt{a})\,a^{r/2}}{\sqrt{a}} = a^{\frac{r}{2}}\left(\frac{1}{\sqrt{a}}-1\right) = \left(\frac{|S|}{|S|+1}\right)^{\frac{r}{2}}\left(\sqrt{\frac{|S|+1}{|S|}}-1\right) \tag{21}$$

Fig. 2(b) plots the probabilities $q_A(r)$, $q_U(r)$, and $q_P(r)$ for instances of the 3-MSTP on $n = 100$ nodes.

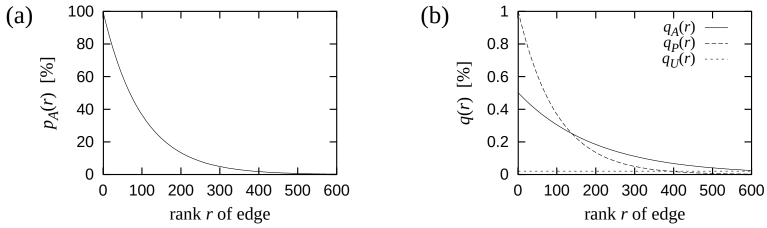

Fig. 2. (a) Approximation of $p(r)$ by $p_A(r) = a^r$ for the 3-MSTP on $n = 100$ nodes, and (b) corresponding edge-selection probabilities $q_A(r)$ (approximately optimal), $q_P(r)$ (proportional to $p_A(r)$), and $q_U(r)$ (uniform).

4 Biasing Mutation for the *d*-MSTP

We consider an EA for the d-MSTP as described in [9]. Mutation is performed by including a random new edge into a feasible solution and removing another randomly chosen edge from the introduced cycle such that the degree constraint is never violated. If a selected edge already appears in the current solution or the degree constraint cannot be met, the selection is repeated. We apply the following strategies for selecting the edge to be included.

UNIF: Each edge is randomly chosen with probability $q_U(r) = 1/m$.

OPTEX: Edges are selected according to the approximately optimal selection probabilities $q_A(r)$ with respect to $EX(e^*)$.

To perform this edge-selection efficiently, 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 (21), we use the inverse of the corresponding cumulative distribution function $F(r)$:

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

$$= 1 - a^{r/2} = 1 - \left(\frac{|S|}{|S| + 1} \right)^{\frac{r}{2}} . \tag{22}$$

The inverse of $F(r)$ is

$$r = \frac{2 \log(1 - F(r))}{\log |S| - \log(|S| + 1)} . \tag{23}$$

$\mathcal{R}$ can be calculated from $\mathcal{U}$ by setting $F(r) = \mathcal{U}$ in (23) and rounding:

$$\mathcal{R} = \left\lfloor \frac{2 \log(1 - \mathcal{U})}{\log |S| - \log(|S| + 1)} \right\rfloor \bmod m + 1 . \tag{24}$$

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

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

$$F(r) = \sum_{i=1}^{r} \frac{a^i}{|S|} = 1 - \left(\frac{|S|}{|S|+1}\right)^r . \tag{25}$$

This yields

$$\mathcal{R} = \left\lfloor \frac{\log(1-\mathcal{U})}{\log|S| - \log(|S|+1)} \right\rfloor \bmod m + 1 . \tag{26}$$

$\mathbf{N}_{\boldsymbol{\beta}}$**:** This edge-selection strategy is based on normal distributions as proposed in [9]. The rank of a selected edge is

$$\mathcal{R} = \lfloor |\mathcal{N} \cdot \beta \cdot n| \rfloor \bmod m + 1 , \tag{27}$$

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

INVW: Each edge $e \in E$ is selected according to probabilities inversely proportional to the edge weights $w(e)$. Such a technique was used in [3] for choosing edges during recombination for the TSP.

5 Experiments on the *d*-MSTP

The five mutation operators were compared in a steady-state EA for the 3-MSTP as described in [9]. The algorithm represents candidate solutions as sets of their edges. Feasible initial solutions are created by a random spanning tree algorithm based on Kruskal's MST algorithm. A new feasible offspring is always derived by performing edge-crossover and mutation. Edge-crossover is based on a random spanning tree algorithm applied to the united edge-sets of two parents. In contrast to [9], no heuristics are used during initialization and recombination. Parents for crossover are selected in binary tournaments with replacement. Each offspring replaces the worst solution in the population except when it duplicates an existing solution.

In the experiments, we considered 50 randomly created Euclidean instances of each size $n = 50$, 100, and 200. The population size was $2\,n$, and the EA terminated if an optimal solution (determined by branch-and-cut) had been reached or the number of evaluations exceeded $5000\,n$.

Runs were performed on each instance with each mutation. For normal-distribution-based edge-selection N_β, β was set to 0.75, 1, 1.5, 2, and 3. Table 2 shows, for each size n and each operator, the percentage of runs that identified optimal solutions and the average number of evaluations in these runs. The best values are printed in bold.

Table 2. Results of the EA for the 3-MSTP with each mutation on Euclidean instances of size $n = 50$, 100, and 200: percentages of runs that found optimal solutions (*%-hits*) and average numbers of evaluations of those runs in thousands (*eval*/1000).

n		UNIF	OPTEX	PROPP	$N_{0.75}$	N_1	$N_{1.5}$	N_2	N_3	INVW
50	*%-hits*	98	**100**	**100**	48	68	94	**100**	98	98
	eval/1000	54.0	11.7	16.9	40.9	25.0	**10.6**	15.0	11.4	20.4
100	*%-hits*	66	**100**	88	20	46	82	90	94	86
	eval/1000	288.2	49.1	67.0	160.8	96.1	93.8	**46.8**	61.7	115.4
200	*%-hits*	8	**96**	78	6	46	64	76	66	46
	eval/1000	887.5	198.7	246.1	452.3	382.0	223.9	216.9	**195.4**	618.6

OPTEX performed best on all three sizes; it found optimal solutions on nearly all the instances, and its average numbers of evaluations are among the lowest. Those mutations with lower numbers of evaluations exhibit significantly poorer hit rates. UNIF needed on average the most evaluations, followed by INVW and $N_{0.75}$. Experiments on uniform instances showed similar tendencies.

6 Biased Mutation for the TSP

In comparison to the d-MSTP, incorporating biased edge-selection techniques into mutation of an EA for the TSP is more difficult. A commonly used mutation operator for the TSP acting on a permutation representation is inversion.

This operator can be modified to include a specific new edge selected by one of the above strategies: We invert the substring beginning after the selected edge's first node and ending with the selected edge's second node.

Note, however, that in addition to the selected edge, a second new edge is implicitly included. This second edge depends on the first edge and the current tour; it cannot be chosen according to the edge-selection strategy.

This side-effect strongly influences the idea of biased mutation and is expected to affect performance. Experiments with the TSP similar to those with the d-MSTP did not show significant differences among the edge-selection methods.

7 Conclusions

The rank-based probabilities with which edges appear in optimal solutions of Euclidean and uniform instances of the d-MSTP and the TSP were empirically analyzed and approximated by an exponential function. We then 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 finding an edge of an optimal solution is minimized.

Using the degree-constrained minimum spanning tree problem, five different edge-selection strategies for mutation were described and compared. With the scheme using the approximately optimal probabilities $q_A(r)$, the EA identified optimal solutions most often and with comparatively few iterations.

On the traveling salesman problem, however, mutation that introduces one edge always introduces a second as well. While the first may be chosen according to certain probabilities, the second depends on the first and on the current tour. This side-effect overwhelms the differences between the various mutation operators. This analysis nonetheless suggests that mutation that includes edges according to probabilities derived in the proposed way might be effective in EAs for graph problems in which the introduction of one edge does not require the inclusion of others.

References

1. N. Deo and P. Micikevicius. A heuristic for a leaf constrained minimum spanning tree problem. *Congressus Numerantium*, 141:61–72, 1999.
2. C. Fonseca, J.-H. Kim, and A. Smith, editors. *Proceedings of the 2000 IEEE Congress on Evolutionary Computation*. IEEE Press, 2000.
3. J. J. Grefenstette. Incorporating problem specific knowledge into genetic algorithms. In L. Davis, editor, *Genetic Algorithms and Simulated Annealing*, pages 42–60. Morgan Kaufmann, 1987.
4. B. A. Julstrom. Very greedy crossover in a genetic algorithm for the Traveling Salesman Problem. In K. M. George, J. H. Carroll, E. Deaton, D. Oppenheim, and J. Hightower, editors, *Proceedings of the 1995 ACM Symposium on Applied Computing*, pages 324–328. ACM Press, 1995.
5. B. A. Julstrom and G. R. Raidl. Weight-biased edge-crossover in evolutionary algorithms for two graph problems. In G. Lamont, J. Carroll, H. Haddad, D. Morton, G. Papadopoulos, R. Sincovec, and A. Yfantis, editors, *Proceedings of the 16th ACM Symposium on Applied Computing*, pages 321–326. ACM Press, 2001.
6. J. Knowles and D. Corne. A new evolutionary approach to the degree constrained minimum spanning tree problem. *IEEE Transactions on Evolutionary Computation*, 4(2):125–134, 2000.
7. M. Krishnamoorthy, A. T. Ernst, and Y. M. Sharaiha. Comparison of algorithms for the degree constrained minimum spanning tree. Technical report, CSIRO Mathematical and Information Sciences, Clayton, Australia, 1999.
8. I. Ljubic and J. Kratica. A genetic algorithm for the biconnectivity augmentation problem. In Fonseca et al. [2], pages 89–96.
9. G. R. Raidl. An efficient evolutionary algorithm for the degree-constrained minimum spanning tree problem. In Fonseca et al. [2], pages 104–111.
10. G. R. Raidl and C. Drexel. A predecessor coding in an evolutionary algorithm for the capacitated minimum spanning tree problem. In C. Armstrong, editor, *Late Breaking Papers at the 2000 Genetic and Evolutionary Computation Conference*, pages 309–316, Las Vegas, NV, 2000.
11. S. Thienel. *ABACUS – A Branch-And-CUt System.* PhD thesis, University of Cologne, Cologne, Germany, 1995.