
A Permutation-Coded Evolutionary Algorithm for the Bounded-Diameter Minimum Spanning Tree Problem

Bryant A. Julstrom

Department of Computer Science,
St. Cloud State University
St. Cloud, MN, 56301 USA
julstrom@eeyore.stcloudstate.edu

Günther R. Raidl

Institute of Computer Graphics and Algorithms
Vienna University of Technology
1040 Vienna, Austria
raidl@ads.tuwien.ac.at

Abstract

The diameter of a tree is the largest number of edges on any path between two vertices in it. Given a weighted, connected, undirected graph G and a bound $D \geq 2$, the bounded-diameter minimum spanning tree problem seeks a spanning tree on G of minimum weight whose diameter does not exceed D . An evolutionary algorithm for this NP-hard problem encodes candidate trees as permutations of their vertices. The first vertex (if D is even) or the first two vertices (if D is odd) form the center of the tree a permutation represents. A greedy heuristic appends the remaining vertices to the tree in their listed order, as economically as possible, while maintaining the diameter bound. In tests on 25 Euclidean problem instances, this EA identifies shorter trees on average than does an EA that encodes trees as sets of their edges, though it takes longer.

1 Introduction

The primary determinant of the effectiveness of an evolutionary algorithm (EA) is the interaction between the coding by which chromosomes represent candidate solutions and the variation operators that the EA applies to those chromosomes. When a greedy heuristic exists for a problem, an EA can encode solutions as permutations of the problem's elements. The heuristic builds the solution a permutation represents by appending the problem's elements to the solution in the order the permutation specifies. The algorithm searches the problem's solution space by searching the space of permutations of its elements, and that search is aided by the greediness of the heuristic.

Permutation-coded EAs have been applied to a variety of problems, including partitioning problems (Jones and Beltramo, 1991), the minimum-span frequency assignment problem (Valenzuela et al., 1998), the traveling salesman problem (Julstrom, 1999), set covering problems (Aickelin, 2001), and the placement of cellular transmitters (Raisanen and Whitaker, 2003), among others. We present such an algorithm for the bounded-diameter minimum spanning tree problem.

Let T be a spanning tree on a collection of vertices. The *eccentricity* of a vertex v is the maximum number of edges on a path from v to any other vertex in T . The *diameter* of T is the maximum eccentricity of its vertices, thus the largest number of edges on any path in T . The *center* of T is the one vertex (if T 's diameter is even) or the two adjacent vertices (if T 's diameter is odd) of minimum eccentricity.

Given a connected, undirected graph G and an integer bound $D \geq 2$, a *bounded-diameter spanning tree* (BDST) is a spanning tree on G whose diameter does not exceed D . When numerical weights label G 's edges, the weight of a spanning tree is the sum of its edges' weights, and a *bounded-diameter minimum spanning tree* (BDMST) is a BDST on G of minimum weight. The search for a BDMST is NP-hard for $4 \leq D < n - 1$ (Garey and Johnson, 1979, p.206).

Abdalla et al. (2000) described a greedy heuristic, based on Prim's MST algorithm, for the BDMST problem. An improved greedy heuristic also imitates Prim but begins at (or in) the center of the tree it builds (Raidl and Julstrom, 2003b). An evolutionary algorithm for the problem encodes candidate trees as sets of their edges augmented with their center vertices. It identifies shorter trees than does either greedy heuristic (Raidl and Julstrom, 2003b).

We present here an EA for the BDMST problem that encodes trees as permutations of the graph's vertices. The center-based greedy heuristic identifies the BDSTs

these chromosomes represent. In tests on 25 Euclidean instances of the BDMST problem, the permutation-coded EA finds shorter trees than does the edge-set-coded EA, though it takes longer to do so.

The following sections of this paper describe the greedy heuristics for the BDMST problem, the edge-set-coded EA, the permutation-coded EA, and the comparisons of the two evolutionary algorithms on the test problem instances.

2 Greedy Heuristics for the BDMST Problem

The heuristic of Abdalla et al. (2000) is called One Time Tree Construction (OTTC); it imitates Prim’s algorithm (Prim, 1957). Prim builds an unconstrained minimum spanning tree from a start vertex by repeatedly appending the lowest-weight edge between a vertex in the tree and one not yet connected to it. The diameter of the tree at each step can always be found. Thus OTTC begins from an arbitrarily chosen start vertex and repeatedly extends the tree with the lowest-weight edge to an unconnected vertex whose inclusion does not violate the diameter bound. Note that the resulting tree depends on the vertex chosen as start vertex.

OTTC keeps track of the eccentricities of and path lengths between all the vertices in the tree. Appending each new edge changes some of these values, so that in the worst case updating the algorithm’s data structures requires time that is quadratic in the number n of vertices. $n - 1$ edges are included in the tree, of course, so the algorithm’s time is $O(n^3)$.

This time can be reduced by a factor of n if the start vertex is initially fixed as the center (if the diameter bound D is even) or as one of the two center vertices (if D is odd) of the spanning tree (Raidl and Julstrom, 2003b). In this case, the heuristic must maintain only the depth of each vertex: the length of the path from it to the center. This value is fixed when each vertex joins the tree, and no vertex may have a depth greater than $\lfloor D/2 \rfloor$. The sketch in Figure 1 summarizes the center-based heuristic.

Each next vertex can be chosen greedily, at random, or by any other heuristic. In any case, it is connected to the tree by the lowest-weight edge whose inclusion does not yield a tree of diameter greater than D . Repeated trials of the randomized version on a variety of Euclidean instances returned much lower-weight BDMSTs than did OTTC (Raidl and Julstrom, 2003b).

```

T ← ∅
vo ← start vertex
U ← V - {vo}
C ← {vo}
depth[vo] ← 0
if D is odd
    v1 ← a vertex in U
    T ← {(vo, v1)}
    U ← U - {v1}
    C ← C ∪ {v1}
    depth[v1] ← 0
while C ≠ V do
    u ← a vertex in U
    v ← the vertex in C nearest u
    T ← T ∪ {(u, v)}
    U ← U - {u}
    depth[u] ← depth[v] + 1
    if depth[u] < ⌊D/2⌋
        C ← C ∪ {u}
return T

```

Figure 1: The center-based heuristic for the bounded-diameter minimum spanning tree problem. T is the tree’s edge set, V is the graph’s vertex set, U is the set of unconnected vertices, and C is the set of connected vertices to which a new edge can be connected without violating the diameter bound.

3 An Edge-Set-Coded Evolutionary Algorithm

A general study of the edge-set representation for spanning tree problems can be found in Raidl and Julstrom (2003a). We have previously described an evolutionary algorithm for the BDMST problem that encodes candidate trees directly as sets of their edges, each augmented with its center vertex or vertices (Raidl and Julstrom, 2003b). A chromosome’s fitness is the total weight of its tree, which can be found, by scanning the edge-set, in time that is $O(n)$.

The algorithm’s operators generate only valid spanning trees; that is, trees whose diameter does not exceed the bound D . The randomized center-based heuristic described in Section 2 provides the spanning trees for the EA’s initial population. The recombination operator builds one offspring tree from two parents. It begins by selecting one or two center vertices from the parents’ centers. It extends the tree, while conforming to the diameter bound, with edges common to both parents, if possible, and with edges found in one parent if necessary. An efficient implementation

of the operator builds temporary sets of adjacency lists to represent the parent trees. Its time is $O(n)$.

The EA applies four mutation operators. The first removes a random edge from the parent tree and rebuilds the tree from its unchanged center, as in the recombination operator. It uses parental edges when possible and other feasible edges chosen at random when it must. The second shifts the parent tree’s center to (include) an adjacent vertex and rebuilds the tree as in the first mutation. The third removes a random edge from the parent tree, thus disconnecting a subtree rooted at a vertex r . It reconnects the tree with the lowest-weight edge whose inclusion does not violate the diameter bound.

The fourth mutation operator optimally rearranges a subtree of the parent tree. It randomly selects a vertex at depth $\lfloor D/2 \rfloor - 1$ in the parent tree and tries each vertex in that subtree as its root, with a direct edge to each remaining vertex. It implements in the offspring the rearrangement of lowest weight.

The first three mutations require times that are linear in n . In the last mutation, let s be the number of vertices in the subtree. Building and evaluating a subtree requires time that is $O(s)$, and there are s possible configurations to investigate, since each vertex in the subtree must be considered as its root. Thus the operator’s total time is $O(n + s^2)$. Note that usually $s \ll n$.

The augmented edge-set coding and these operators were applied in a steady-state evolutionary algorithm. The EA selects parents in tournaments with replacement. It applies recombination with a probability for each offspring of $p < 1.0$, and it always applies exactly one of the mutation operators. Each offspring replaces the population’s worst solution, except that duplicates are discarded.

4 A Permutation-Coded Evolutionary Algorithm

We propose an evolutionary algorithm for the bounded-diameter minimum spanning tree problem that encodes candidate spanning trees as permutations of their vertices. The first vertex listed (if D is even) or the first two vertices (if D is odd) form the center of the spanning tree a permutation represents. The center-based greedy heuristic of Section 2 completes the tree by appending the remaining vertices in their listed order. The resulting algorithm is thus a hybrid of the greedy heuristic used to decode permutations and an evolutionary algorithm that searches the space

of permutations. Every permutation represents a valid bounded-diameter spanning tree via the greedy heuristic; the heuristic enforces the diameter bound.

The chromosomes of the algorithm’s initial population are random permutations; generating one requires time that is $O(n)$. Evaluating a chromosome requires identifying the spanning tree it represents—that is, applying the greedy heuristic—then summing the weights of the tree’s edges. The time of the latter step is linear in n , but that of the former is quadratic. Thus, the time of evaluation is $O(n^2)$.

Each vertex’s position in a chromosome determines when the greedy heuristic includes it in the spanning tree the chromosome represents, so the EA applies Partially Mapped Crossover (PMX) (Goldberg and Robert Lingle, 1985), which tends to preserve the positions of symbols from parents to offspring. PMX generates one offspring from two parents. It chooses a random segment within one parent and swaps each symbol in this segment with the symbol appearing at the same position in the second parent. The offspring permutation inherits the selected segment exactly from the second parent. It inherits the remaining positions primarily from the first parent. In the implementation of this operator, an auxiliary array holds each vertex’s position in the offspring chromosome. This array allows PMX to find a vertex from the target segment in constant time, so the operator’s time is $O(n)$.

Mutation swaps the vertices at two random positions in the parent chromosome, thus exchanging the times at which the heuristic includes those vertices in the spanning tree. Three assignments swap the two vertices, but copying the parent chromosome into the offspring takes time that is $O(n)$.

The structure of the permutation-coded EA is identical to that of the edge-set-coded algorithm of the previous section. It is steady-state, selects parent chromosomes in tournaments with replacement, applies recombination sometimes and mutation always, and replaces the population’s worst chromosome with each offspring, except when the offspring duplicates a chromosome already in the population.

5 Tests

The edge-set-coded EA of Section 3 and the permutation-coded EA of Section 4 were compared on 25 Euclidean instances of the bounded-diameter minimum spanning tree problem. There were five instances each of $n = 50, 70, 100, 250,$ and 500 points in the

Table 1: For each set of 50 trials of the edge-set-coded EA and the permutation-coded EA on each of the 25 BDMST instances: the length of the shortest tree found, the average and standard deviation of the 50 tree lengths, and the average number of iterations in the trials.

Instance			Edge-Set-Coded EA				Permutation-Coded EA			
n	D	num.	shortest	mean	stddev	iterations	shortest	mean	stddev	iterations
50	5	1	7.60	7.93	0.22	33 947	7.60	7.78	0.16	36 926
50	5	2	7.68	7.87	0.14	36 403	7.75	7.84	0.10	27 955
50	5	3	7.24	7.51	0.15	27 919	7.25	7.41	0.14	31 725
50	5	4	6.59	6.75	0.15	31 382	6.62	6.65	0.05	40 703
50	5	5	7.32	7.49	0.09	34 924	7.39	7.47	0.08	34 369
70	7	1	7.36	7.66	0.13	82 351	7.23	7.32	0.06	68 863
70	7	2	7.26	7.57	0.14	65 581	7.12	7.21	0.06	61 106
70	7	3	7.18	7.49	0.15	79 838	6.99	7.13	0.11	64 824
70	7	4	7.68	8.01	0.17	56 401	7.53	7.62	0.07	54 316
70	7	5	7.45	7.72	0.16	83 404	7.27	7.32	0.06	71 344
100	10	1	8.00	8.30	0.12	189 026	7.77	7.89	0.06	88 831
100	10	2	8.10	8.41	0.16	205 891	7.88	8.02	0.08	82 560
100	10	3	8.22	8.61	0.19	176 043	7.93	8.11	0.10	88 467
100	10	4	8.27	8.57	0.17	163 142	8.00	8.13	0.07	92 993
100	10	5	8.48	8.72	0.15	164 651	8.16	8.30	0.07	89 735
250	15	1	12.93	13.36	0.19	471 803	12.41	12.52	0.07	278 984
250	15	2	12.86	13.25	0.20	466 047	12.20	12.37	0.09	311 349
250	15	3	12.69	13.06	0.20	464 618	12.12	12.23	0.08	309 699
250	15	4	13.22	13.65	0.19	442 446	12.64	12.77	0.08	305 125
250	15	5	13.02	13.40	0.19	497 450	12.37	12.56	0.08	316 006
500	20	1	18.33	18.77	0.29	527 659	17.10	17.35	0.12	712 011
500	20	2	18.17	18.60	0.19	652 009	16.97	17.20	0.14	736 921
500	20	3	18.33	18.76	0.28	504 315	17.13	17.28	0.10	746 259
500	20	4	18.32	18.74	0.18	654 871	17.09	17.29	0.12	742 231
500	20	5	17.80	18.40	0.28	648 148	16.79	17.03	0.15	705 313

unit square. They are found in Beasley’s OR-Library¹ (Beasley, 1990), listed as instances of the Euclidean Steiner problem. The library contains fifteen instances of each size; we have used the first five.

The points are treated as the vertices of complete graphs whose edge weights are the Euclidean distances between the points. For $n = 50$, the diameter bound D is 5; when $n = 70$, $D = 7$; when $n = 100$, $D = 10$; when $n = 250$, $D = 15$, and when $n = 500$, $D = 20$.

For the tests, the two EA’s parameters were set identically. Both algorithms’ populations contained 400 chromosomes. Both selected parent chromosomes in tournaments of size three. Both applied recombination with a probability of 60%; the edge-set-coded EA applied its edge-delete and center-move mutations (the first two listed) with probabilities of 20% and the remaining two mutations with probabilities of 30%. Both EAs terminated when their populations’ best

chromosomes had not improved for 100 000 new chromosomes.

On each instance, both EAs were run 50 independent times. Table 1 summarizes the results of these sets of trials. For each algorithm and each instance it lists the length of the shortest bounded-diameter tree found, the average and standard deviation of the 50 tree lengths, and the average number of iterations in each trial until its best tree.

On the smallest instances, with $n = 50$, the results are ambiguous. On each of the five instances, the edge-set-coded EA identifies in its best run the overall shortest tree (with one tie), while the permutation-coded algorithm consistently returns shorter trees on average. The two algorithms execute approximately the same numbers of iterations to identify their shortest trees.

On all the larger instances, the permutation-coded EA enjoys a decisive advantage. On each instance this EA identifies in its best run the overall shortest tree, and

¹mscmga.ms.ic.ac.uk/info.html

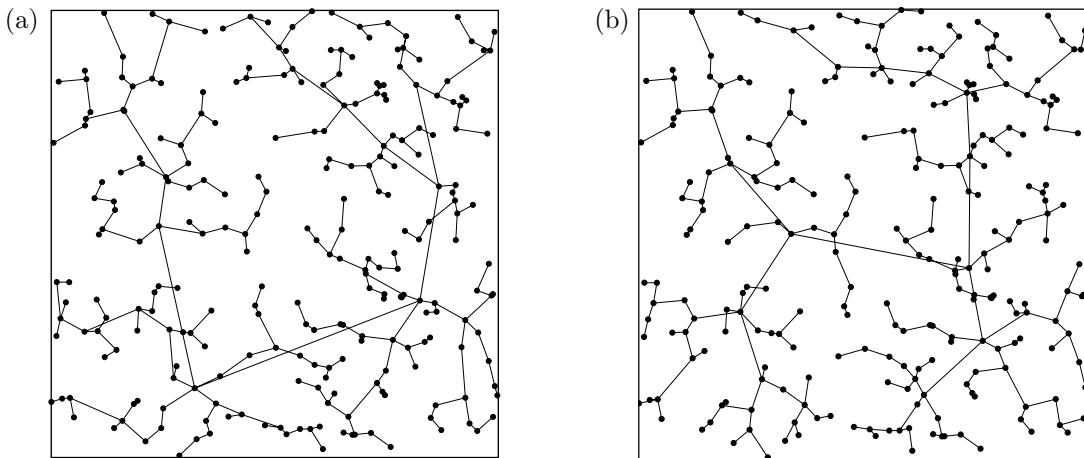


Figure 2: The lowest-weight bounded-diameter spanning trees found by the two EAs on the second problem instance with $n = 250$ and $D = 15$: (a) by the edge-set-coded EA; weight = 12.86; (b) by the permutation-coded EA; weight = 12.20.

its trees are also on average shorter than those found by the edge-set-coded algorithm. Indeed, the average length of the permutation-coded EA’s 50 trees is always less than the length of the shortest tree found by the edge-set-coded EA. Further, the number of iterations the permutation-coded EA requires is on average less than the number expended by the edge-set-coded EA except, interestingly, on the instances with $n = 500$.

Figure 2 shows the shortest trees found by the two algorithms on the second instance with $n = 250$ vertices. The edge-set-coded EA’s tree has length 12.86, and the permutation-coded EA’s tree has length 12.20.

The permutation-coded EA’s superior performance is explained by the greediness of the heuristic that decodes its chromosomes. By appending each vertex as economically as possible (without violating the diameter bound), it aggressively seeks short trees. Since it appends vertices only in the order a permutation dictates, it can still construct trees in which longer edges connect more widely separated groups of vertices.

The permutation-coded EA’s effectiveness, however, comes at a price. In the edge-set-coded EA, evaluation, recombination, and mutation all require times that are at worst slightly larger than linear in n . In the permutation-coded EA, recombination and mutation are $O(n)$, but evaluation’s time is $O(n^2)$. Even though the permutation-coded EA performs fewer evaluations on average than does the edge-set-coded EA, it takes longer, and the difference grows with n .

We compared the two representations and their operators in similar EA structures. Because the

permutation-coded EA takes so much longer to run, a fairer comparison would allow the edge-set-coded EA more time by relaxing its termination condition, using a larger population size, or applying some additional diversity-preservation mechanism. We expect that such modifications would significantly improve the edge-set-coded EA’s solutions.

6 Conclusion

Given a connected, weighted, undirected graph G and a bound $D \geq 2$, the bounded-diameter minimum spanning tree problem seeks a spanning tree on G of smallest weight in which the longest path between any two vertices does not contain more than D edges. Greedy heuristics for this problem imitate Prim’s algorithm. They attach each new vertex with the edge of lowest weight whose inclusion does not yield a tree of diameter greater than D .

A previously described evolutionary algorithm for this problem encodes spanning trees directly as sets of their edges. Its operators build spanning trees with diameters no greater than D from their centers.

A new EA encodes candidate spanning trees as permutations of their vertices. The bounded-diameter spanning tree that a permutation represents is identified by a greedy heuristic. The heuristic uses a permutation’s first vertex (if D is even) or two vertices (if D is odd) as its tree’s center, then appends the remaining vertices in their listed order. It attaches each vertex with the lowest-weight edge to a vertex in the tree whose inclusion does not violate the diameter bound. This EA searches the space of diameter-bounded spanning

trees by searching the space of permutations of the vertices.

The two EAs, with similar structures and parameters, were compared on 25 Euclidean instances of the bounded-diameter minimum spanning tree problem of $n = 50$ to 500 vertices. On the five smallest instances, the EAs' performances were similar, but on the instances with 70 or more vertices, the permutation-coded EA consistently identified shorter bounded-diameter spanning trees in fewer iterations than did the edge-set-coded EA. However, because the heuristic that decodes permutations requires time that is $O(n^2)$, the permutation-coded EA is slower, and its disadvantage in time increases with the size of the problem instances.

References

- A. Abdalla, N. Deo, and P. Gupta. Random-tree diameter and the diameter constrained MST. *Congressus Numerantium*, 144:161–182, 2000.
- U. Aickelin. An indirect genetic algorithm for set covering problems. *Journal of the Operational Research Society*, 53(10):1118–1126, 2001.
- J. E. Beasley. OR-library: Distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41:1069–1072, 1990.
- M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, 1979.
- D. E. Goldberg and J. Robert Lingle. Alleles, loci, and the traveling salesman problem. In J. J. Greffentette, editor, *Proceedings of the First International Conference on Genetic Algorithms*, pages 154–159, Hillsdale, NJ, 1985. Lawrence Erlbaum.
- D. R. Jones and M. A. Beltramo. Solving partitioning problems with genetic algorithms. In R. K. Belew and L. B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 442–449, San Mateo, CA, 1991. Morgan Kaufmann Publishers.
- B. A. Julstrom. Coding TSP tours as permutations via an insertion heuristic. In J. Carroll, H. Haddad, D. Oppenheim, B. Bryant, and G. B. Lamont, editors, *Applied Computing 1999: Proceedings of the 1999 ACM Symposium on Applied Computing*, pages 297–301, New York, 1999. ACM Press.
- G. Lamont, H. Haddad, G. A. Papadopoulos, B. Panda, J. Carroll, R. Stansifer, W. Jones, R. Menezes, and W. Shoaff, editors. *Proceedings of the 2003 ACM Symposium on Applied Computing*, 2003. ACM Press.
- R. C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957.
- G. R. Raidl and B. A. Julstrom. Edge-sets: An effective evolutionary coding of spanning trees. *IEEE Transactions on Evolutionary Computation*, 2003a. To appear.
- G. R. Raidl and B. A. Julstrom. Greedy heuristics and an evolutionary algorithm for the bounded-diameter minimum spanning tree problem. In Lamont et al. (2003), pages 747–752.
- L. Raisanen and R. M. Whitaker. Multi-objective optimization in area coverage problems for cellular communications networks: Evaluation of an elitist evolutionary strategy. In Lamont et al. (2003), pages 714–720.
- C. Valenzuela, S. Hurley, and D. Smith. A permutation based genetic algorithm for minimum span frequency assignment. In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature – PPSN V*, volume 1498 of *LNCS*, pages 907–916, Berlin, 1998. Springer-Verlag.