Multiobjective EA Approach for Improved Quality of Solutions for Spanning Tree Problem

Rajeev Kumar, P. K. Singh, and P. P. Chakrabarti

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur, WB 721 302, India
{rkumar,pksingh,ppchak}@cse.iitkgp.ernet.in

Abstract. The problem of computing spanning trees along with specific constraints is mostly NP-hard. Many approximation and stochastic algorithms which yield a single solution, have been proposed. In this paper, we formulate the generic multi-objective spanning tree (MOST) problem and consider edge-cost and diameter as the two objectives. Since the problem is hard, and the Pareto-front is unknown, the main issue in such problem-instances is how to assess the convergence. We use a multiobjective evolutionary algorithm (MOEA) that produces diverse solutions without needing a priori knowledge of the solution space, and generate solutions from multiple tribes in order to assess movement of the solution front. Since no experimental results are available for MOST, we consider three well known diameter-constrained minimum spanning tree (dc-MST) algorithms including randomized greedy heuristics (RGH) which represents the current state of the art on the dc-MST, and modify them to yield a (near-) optimal solution-fronts. We quantify the obtained solution fronts for comparison. We observe that MOEA provides superior solutions in the entire-range of the Pareto-front, which none of the existing algorithms could individually do.

1 Introduction

Computing a minimum spanning tree (MST) from a connected graph is a well-studied problem and many fast algorithms and analytical analyses are available [1–8]. However, many real-life network optimization problems require the spanning tree to satisfy additional constraints along with minimum edge-cost. For example, communication network design problem for multicast routing of multimedia communication requires constructing a minimal cost spanning/Steiner tree with given constraints on diameter. VLSI circuit design problems aim at finding minimum cost spanning/Steiner trees given delay bound constraints on source-sink connections. Analogously, there exists the problem of degree/diameter-constrained minimum cost networks in many other engineering applications too (see [3] and the references therein).

Many such MST problem instances having a bound on the degree, a bound on the diameter, capacitated trees or bounds for two parameters to be satisfied simultaneously are listed in [3]. Finding spanning trees of sufficient generality and of minimal cost subject to satisfaction of additional constraints is often NP-hard [3, 4]. Many such design problems have been attempted and approximate solutions obtained using heuristics.

For example, the research groups of Deo et al. [5–8] and Ravi et al. [3, 4] have presented approximation algorithms by optimizing one criterion subject to a budget on the other. In recent years, evolutionary algorithms (EAs) have emerged as powerful tools to approximate solutions of such NP-hard problems. For example, Raidl & Julstrom [9, 10] and Knowles & Corne [11, 12] attempted to solve diameter and degree constrained minimum spanning tree problems, respectively using EAs. All such approximation and evolutionary algorithms yield a *single* optimized solution subject to satisfaction of the constraint(s). Moreover, researchers have demonstrated superiority of one algorithm over other algorithms for a *particular* value of a constraint and did not assess the performance over entire range of the values.

We argue that such constrained MST problems are essentially multiobjective in nature. A multiobjective optimizer yields a set of all representative equivalent and diverse solutions; the set of all optimal solutions is the Pareto-front. Secondly, extending this constraint-optimization approach to multi-criteria problems (involving two or more than two objectives/constraints) the techniques require improving upon more than one constraints. Thirdly and more importantly, such approaches may not yield all the representative optimal solutions. For example, most conventional approaches to solve network design problems start with a minimum spanning tree (MST), and thus effectively minimize the cost. With some variations induced by ϵ-constraint method, most other solutions obtained are located near the minimal-cost region of the Pareto-front, and thus do not form the complete (approximated) Pareto-front.

In this work, we try to overcome the disadvantages of conventional techniques and single objective EAs. We use multiobjective EA to obtain a (near-optimal) Pareto-front. For a wide-ranging review, a critical analysis of evolutionary approaches to multiobjective optimization and many implementations of multiobjective EAs, see [13, 14] for algorithms and implementations, and [15] for various applications.

We use Pareto Converging Genetic Algorithm (PCGA) [16] which has been demonstrated to work effectively across complex problems and achieves diversity without needing *a priori* knowledge of the solution space. PCGA excludes any explicit mechanism to preserve diversity and allows a natural selection process to maintain diversity. Thus multiple, equally good solutions to the problem, are provided. Another major challenge to solving unknown problems is how to ensure convergence. Some multiobjective problems have a tendency to get stuck at local Pareto-front [16], therefore, we generate solutions using multiple tribes and merge them to ensure convergence. PCGA assesses convergence to the Pareto-front which, by definition, is unknown in most real search problems of multi-dimensionality, by use of rank-histograms [17]. We consider, without loss of generality, edge-cost and tree-diameter as the two objectives to be minimized, though the framework presented here is generic enough to include any number of objectives to be optimized. Initial results of this work were presented in other conferences [18, 19]. In this paper, we extend the work for larger problem instances, present a systematic approach to assess the convergence, and compare qualitatively and quantitatively the obtained solution-fronts from three well-known techniques, namely, One-Time-Tree Construction (OTTC) [7], Iterative Refinement (IR) [7], and Randomized Greedy Heuristics (RGH) [9] algorithms.

The rest of the paper is organized as follows. In Section 2, we include a brief review of the issues to be addressed for achieving quality solutions in the context of a MOEA. We describe, in Section 3, the representation scheme for the spanning tree and its implementation using PCGA. Then, we present results in Section 4 along with a comparison with other approaches. Finally, we draw conclusions in Section 5.

2 Multiobjective Evolutionary Algorithms : Issues & Challenges

EAs have emerged as powerful black-box optimization tools to approximate solutions for NP-hard combinatorial optimization problems. In the multiobjective scenario, EAs often find effectively a set of mutually competitive solutions without applying much problem-specific information. However, achieving proper diversity in the solution-set while approaching convergence is a challenge in multiobjective optimization, especially for unknown problems.

There exist many algorithms/implementations which have been demonstrated to achieve diverse and equivalent solutions [13, 14]. For diversity, some of the algorithms make explicit use of parameterized sharing, mating restriction and/or some other diversity preserving operator. Apart from its heuristic nature, the selection of the domain in which to perform sharing (variable (genotype) or objective (phenotype)) is also debatable. Any explicit diversity preserving mechanism method needs prior knowledge of many parameters and the efficacy of such a mechanism depends on successful fine-tuning of these parameters. Purshouse & Fleming [20] extensively studied the effect of sharing, along with elitism and ranking, and concluded that while sharing can be beneficial, it can also prove surprisingly ineffective if the parameters are not carefully tuned. Also, it is the experience of almost all researchers that proper tuning of sharing parameters is necessary for effective performance.

In particular to MOST problem where we use a special encoding [10], incorporation of such knowledge is not an easy task. There exist some other MOEAs, e.g., NSGA-II [21] and SPEA2 [22], which have now dispensed away with parameters for explicit niching. However, almost all the multiobjective evolutionary algorithms and implementations have ignored the issue of convergence and use some pre-determined metrics (e..g, number of generational runs) as the stopping criterion. Other common metric used is the distance metric which finds distance of the obtained solution front from the true Pareto front; this is trivially done for known problems. Such a metric is based on a reference and, a true reference is not known for unknown problems. A commonly practiced approach to determine the reference for unknown problems is to extract the reference from the best-solutions obtained so far, and the reference is incrementally updated with every generation in iterative refinement based algorithms.

Kumar & Rockett [17] proposed use of rank-histograms for monitoring convergence of Pareto-front while maintaining diversity without any *explicit* diversity preserving operator. Their algorithm is demonstrated to work for problems of *unknown* nature. Secondly, assessing convergence does not need *a priori* knowledge for monitoring movement of Pareto-front using rank-histograms. Some other studies have been done on combining convergence with diversity. Laumanns et al. [23] proposed an ϵ-dominance for getting an ϵ-approximate Pareto-front for problems whose optimal Pareto-set is *known*.

Many metrics have been proposed for quantitative evaluation of the quality of solutions [13, 14]. Essentially, these metric are divided into two classes:

- Diversity : Coverage and sampling of the obtained solutions across the front, and
- Convergence : Distance of the obtained solution-front from the *reference* front.

Some of the commonly used metrics are R-measure [24], S-measure (hyper-volume) [25], Generational distance (GD) [26], Spread measure [14, 27], and Convergence measure [28]. Some of these metrics (e.g., generational distance, volume of space covered, error ratio measures of closeness of the Pareto-front to the true Pareto front) are only applicable where the solution is known. In case of unknown nature, the metrics are sensitive to the choice of the reference. Other metrics (e.g. ratio of non-dominated individuals, uniform distribution) quantify the Pareto-front and can only be used to assess diversity. Knowles & Corne gave a detailed critical review of these measures in his paper [29], and recommended use of some of the metrics as stable measures. They have also shown the sensitivity of some of the metrics with respect to the arbitrary choice of the reference point/front.

The MOST problem is an NP-hard problem, the actual Pareto-front is not known. In Section 4, we will show that different algorithms give different shapes of the solution front; and interpretation of convergence and diversity from the metrics extracted from different shapes to be done meaningfully is not a straight forward task.

3 Design & Implementation

Evolutionary algorithm operators, namely, mutation and crossover imitate the process of natural evolution, and are instrumental in exploring the search space. The efficiency of the evolutionary search depends how a problem (in this case, a spanning tree) is represented in a chromosome and the reproduction operators are defined. There are many encoding schemes to represent spanning trees – see [10] for a detailed review and comparison. For example, one classic representation scheme is Prüfer encoding which is used by Zhou & Gen [30]. Raidl & Julstrom [10] and Knowles & Corne [12] have pointed out that Prüfer numbers have poor locality and heritability and are thus unsuitable for evolutionary search. Deo et al. suggested use of other variants of Prüfer mappings [8]. Recently, Raidl & Julstrom [10] proposed spanning trees to be represented directly as sets of the edges and have shown locality, heritability and computational efficiency of the edge sets for evolutionary search. (While writing this paper, we have come across a newer encoding scheme [31] for tree-based combinatorial optimization problems, which is shown to give superior performance on larger instances of dc-MST; we are currently using this encoding scheme for MOST and the results will be published elsewhere.) In all the results reported in this paper, we use edge-set scheme for representing spanning trees to exploring the search space.

Initial Population : We generate initial population based on random generation of spanning trees. We do not choose the cheapest edge from the currently eligible list of edges (as per Prim's algorithm) rather we select a random edge from the eligible list; this is done to un-bias the randomly generated population from the links found in MST. The

other variants of generating initial trees could be based on One-Time-Tree Construction (OTTC) [7] and Randomized Greedy Heuristics (RGH) [9] algorithms.

Fitness Evaluation : We use Pareto-rank based EA implementation. The Pareto-rank of each individual is equal to one more than the number of individuals dominating it in the multiobjective vector space. All the non-dominated individuals are assigned rank one. The values of the two objectives to be minimized (cost and diameter) are used to calculate rank of the individual. Based on the two objectives rank of the individual is calculated. In this work, we calculate fitness of an individual by an inverse quadratic function of the Pareto-rank.

Other Genetic Operator : We select crossover operator to provide strong habitability such that the generated trees consist of the parental edges as far as possible. For generating valid trees, we include non-parental edges into the offspring tree. The crossover operator used in this work is a variant of the operator used by Raidl & Julstrom [9]. Raidl & Julstrom used the diameter information to know the center of the tree. Since we do not generate trees for a specific value of a constrained diameter, we do not have the diameter information to be embedded in the crossover. We start with an edge which is common in both parents as the start edge.

The mutation operators used in this work are again the variants of the operators used by Raidl & Julstrom and designed for edge-set encoding [9]. They designed all the four mutation operators based on the diameter information. In our case, we do not know diameter value, therefore, we adapted their mutation operators to work for diameter-independent values.

Ensuring Convergence : We compute *Intra*-island rank-ratio histogram for each epoch of the evolutionary evolution and monitor the movement of the Pareto-front. Since, this is a hard problem, it is likely that the improvement may get trapped in local minima. To ensure a global (near-) optimal Pareto-front, we use a multi-tribal/island approach and monitor the Pareto-front using *Inter*-island rank histogram. Our multi-island/tribal approach is essentially a test on convergence rather parallelizing the computational efforts as done by others, e.g., Cantu-Paz [32]. For details of computation of Intra-island rank-ratio and Inter-island rank histograms, see [16].

Algorithm : The PCGA algorithm [16] used in this work is a steady-state algorithm and can be seen as an example of (μ + 2) – Evolutionary Strategy (ES) in terms of its selection mechanism [13, 14]. In this algorithm, individuals are compared against the total population set according to a tied Pareto-ranking scheme and the population is selectively moved towards convergence by discarding the lowest ranked individuals in each evolution. In doing so, we require no parameters such as size of the sub-population in tournament selection or sharing/niching parameters. Initially, the whole population of size N is ranked and fitness is assigned by interpolating from the best individual (rank = 1) to the lowest (rank $\leq N$) according to some simple monotonic function. A pair of mates is randomly chosen biased in the sizes of the roulette wheel segments and crossed-over and/or mutated to produce offspring. The offspring are inserted into the population set according to their ranks against the whole population and the lowest

ranked two individuals are eliminated to restore the population size to N. The process is iterated until a convergence criterion based on Intra-island rank-ratio and Inter-island rank histogram is achieved [16, 17]. A brief Pseudocode of the PCGA is included in Algorithm 1.

Algorithm 1 : Pareto Converging GA

```
Input: N - size of initial population and GA parameters
Output: a set of (near-) optimal solutions
Algorithm:
Generate an initial population of size N
Compute individual's objective vector
Pareto-rank the population and generate rank-ratio histogram
while Intra-island rank-ratio histogram does not satisfy stopping criterion do
    Select two parents using a selection scheme
    Perform crossover and mutation to generate two offsprings
    Compute objective vectors of offsprings
    Pareto-rank the population including offsprings
    Remove the two least fit individuals to keep the size N
    Generate rank-ratio histogram
end while
One while-loop for Inter-island rank-histogram satisfying stopping criterion
Output set of solutions
```

If two individuals have the same objective vector, we lower the rank of one of the individual by one; this way, we are able to remove the duplicates from the set of non-dominated solutions without loss of generality. For a meaningful comparison of two real numbers during ranking, we restrict the floating-point precision of the objective values to a few units of precision; this is problem dependent and can be tuned by trial-and-error during few initial runs of the algorithm. Otherwise, this algorithm does not explicitly use any other diversity preserving mechanism. However, lowering the rank of the individual having the identical objective vector (with restricted units of precision) is analogous in some way to a sort of sharing/niching mechanism (in objective space) which effectively controls the selection pressure and thus *partly* contributes to diversity (For other factors that contribute to diversity, see [16]).

4 Results

We tested generation of dual objective spanning tree using our MOEA framework and selected benchmark data taken from Beasley's OR library[1]. The OR-Library is a collection of test data sets for a variety of Operations Research (OR) problems. We considered the Euclidean Steiner problem data which was used by previous researchers, e.g., Raidl-SAC. We considered datasets of up to 250 nodes for this work, and few representative results are included in rest of this Section.

[1] http://mscmga.ms.ic.ac.uk/info.html

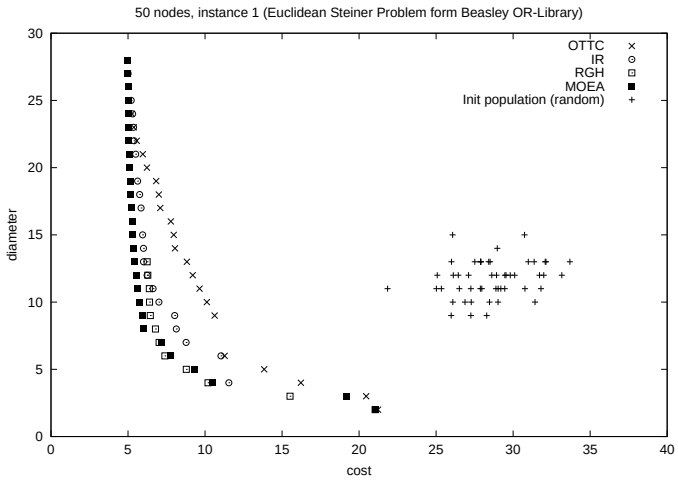

Fig. 1. Pareto front generated from evolutionary algorithm for a 50 node data. Initial population and other fronts generated from OTTC, IR and RGH algorithms are also shown in the plot.

For comparison, we also include results obtained from three well-known diameter constrained algorithms, namely, One-Time Tree Construction (OTTC) [7], Iterative Refinement (IR) [7] and Randomized Greedy Heuristics (RGH) [9] algorithms. All the three algorithms have been demonstrated for Beasley's OR data and few results included in their respective papers. All three algorithms are *single* objective single constraint algorithms and generate a single tree subject to the diameter constraint. Our MOST algorithm simultaneously optimizes both the objectives and generates a (near-optimal) Pareto-front which comprises a set of solutions. Therefore, we iteratively run all the three - OTTC, IR and RGH - algorithms by varying the value of the diameter constraint and generate sets of solutions to form the respective Pareto-fronts, for comparison with the Pareto-front obtained from the proposed multiobjective evolutionary algorithm. For randomized algorithms, evolutionary and RGH, we have repeated experiments ten times to observe the variability due to randomization, and include here a single set of representative results obtained from the runs.

First, we include results obtained for 50 node data from all the four - OTTC, IR, RGH and our proposed MOEA - algorithms in Fig. 1. Initial population for the proposed MOEA is also shown in Fig. 1, and the corresponding intra rank-ratio histogram is shown in Fig. 2. The rank-ratio histogram after one iteration (epoch) is included in Fig. 3. The reduction in the size of the tail of the histogram indicates movement of the Pareto-front towards convergence which is substantial in this case. At convergence, the final rank-ratio histogram is depicted in Fig. 4. At this stage, all the entries are non-dominated and status of none of the individual was changed from non-dominated to dominated one in the past iteration/epoch. However, this necessarily does not mean that the Pareto-front as shown in Fig. 1 is necessarily optimal. Second, we also do not have *a priori* knowledge of the solution space, therefore, we are not in any position

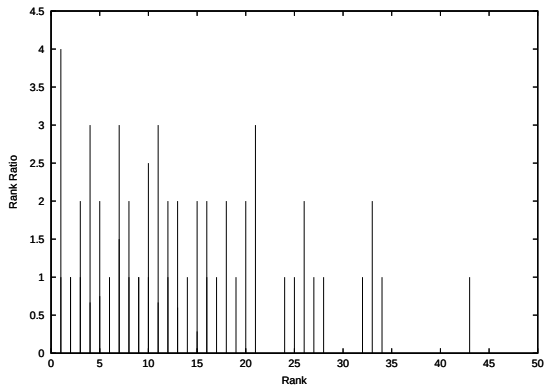

Fig. 2. Rank-ratio histogram computed from the initial population for 50 node data.

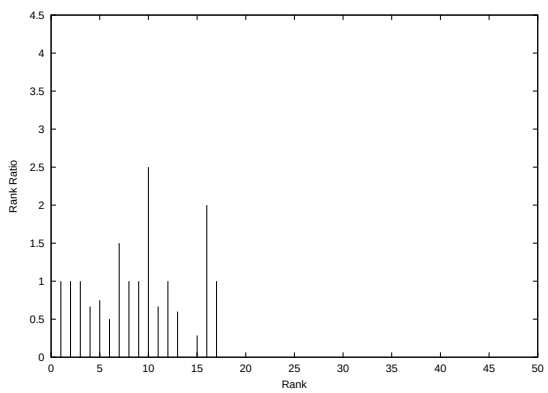

Fig. 3. Rank-ratio histogram computed from the population after first iteration/epoch for 50 node data. Movement of the solution-front can easily be seen with reduction of the histogram tail in comparison with Fig. 2

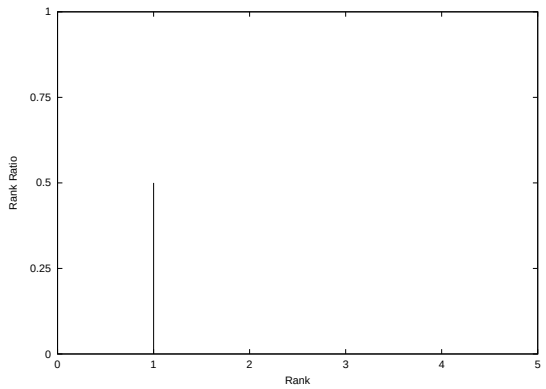

Fig. 4. Rank-ratio histogram at convergence for 50 node data.

to know about the distance between the actual Pareto-front and the obtained solution-front. Therefore, we run MOEA for another run with another set of randomly initialized population, and get a solution set which was marginally superior to the previous one. On merging these two sets, we could get little improvement in the Pareto-front. We may also run MOEA for the third run, and we may again get improvement. The movement of the Pareto-front with each additional run can be monitored on Inter-island rank-histogram (Fig. 7).

Next, we experimented for 100 node data. Results obtained from all the four algorithms are included in Fig. 5. The solutions obtained by MOEA are improved by running the algorithm again, and merging the obtained solutions to form a single Pareto-front. Results obtained from two randomly initialized runs of evolutionary algorithm for 100 node data to form an improved Pareto-front are included in Fig. 6. It can be seen from Fig. 6 that the solutions are improved in the lower and higher ranges of diameters. We plot the Inter-tribal rank-histogram for these two runs of the algorithms and include in Fig. 7. The movement of the solution front by merging the second set of solutions can easily be seen by the long-tail of the rank-histogram in Fig. 7. Otherwise, in a converged state, the rank-histogram should ideally have a single peak of normalized value one at rank one. This indicates that the solution quality is marginally improved by merging two tribes. This is possible because some of the solution points obtained from two tribes of MOEA were distinct and diverse in lower and higher ranges of diameter. This is the clear advantage of using the multi-tribal approach; the results could still be improved with a few more tribes. Such a multi-island/tribal approach is a test on convergence too.

We also collected results from 250 node data; results are included in Fig. 8. Solutions obtained from MOEA are marginally sub-optimal compared to RGH algorithm in very low-range of diameter; this is obvious because MOEA is generic for any diameter values while RGH is tuned to the specific values. The quality of solutions can be further improved by merging solutions obtained from few more tribes, and having a test on convergence. Moreover, if genetic operators of MOEA are tuned to a specific value of diameter like RGH, it will give superior solutions.

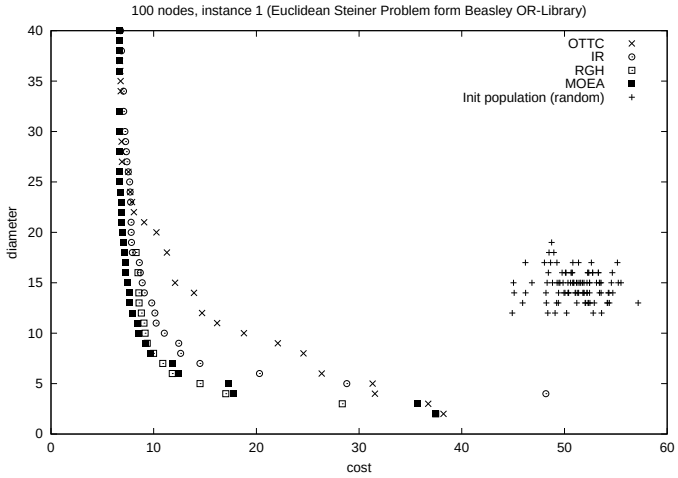

Fig. 5. Pareto front generated from evolutionary algorithm for a 100 node data. Initial population is also shown. Other fronts from OTTC, IR and RGH algorithms are also shown in the plot.

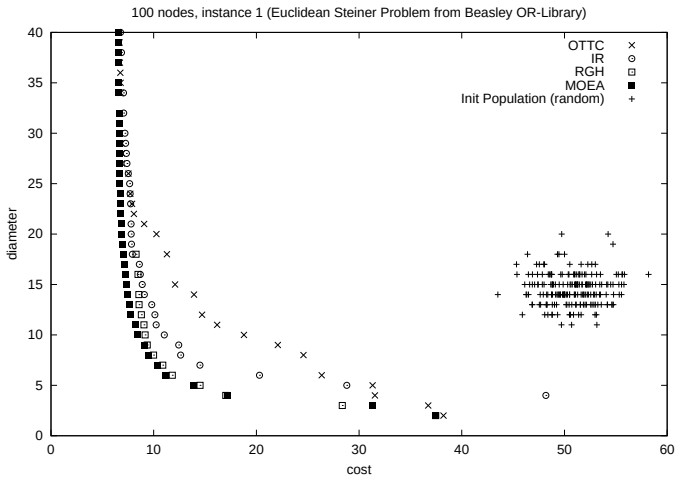

Fig. 6. Improved Pareto front generated from two tribes of evolutionary algorithm for the 100 node data; improvement in the lower and higher ranges of diameters are clearly visible. Initial population of one single tribe is shown. Other fronts from OTTC, IR and RGH algorithms are also shown in the plot.

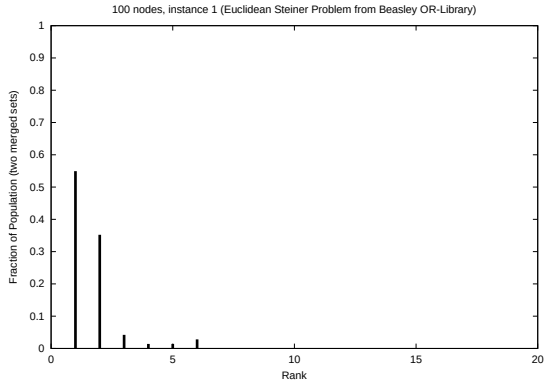

Fig. 7. Inter-Island Rank-histogram computed from two independent runs of the population for 100 node data. A peak value of one located at rank one indicates a no-movement state. In this case, a shift in the Pareto-front is indicated by non-zero population at rank higher than one.

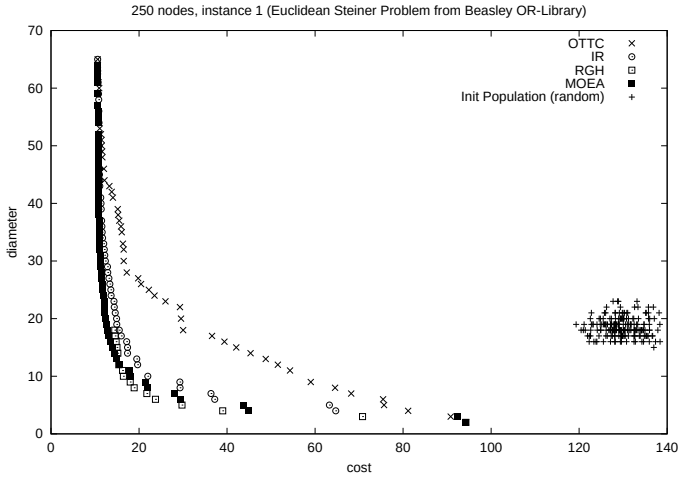

Fig. 8. Pareto front generated from evolutionary algorithm for a 250 node data. Initial population is also shown. Other fronts from OTTC, IR and RGH algorithms are also shown in the plot. Results could be improved by adding solutions from few more tribes.

Finally, we quantitatively evaluate the solution fronts obtained from each of the algorithms. We compute R-measure [24], S-measure [25], Spread [14] and Convergence measures [28], and include representative results for 100 node dataset in Table 1.

Table 1. Diversity and Convergence metrics for 100 node dataset

Algorithm	R-measure	S-measure	Spread	Convergence
OTTC	0.961	8788	0.489	0.072
IR	0.971	8730	0.818	0.050
RGH	0.977	8819	0.864	0.021
MOEA - 1 tribe	0.979	8950	0.769	0.011
MOEA - 2 tribes	0.980	8960	0.671	0.003

It can be observed from Figures 1, 5, 6 and 8, that this is indeed difficult to find the solutions in the higher range of diameter. In fact, RGH algorithm could not find any solution in higher range of diameter; we generated multiple sets of solutions with multiple runs of RGH algorithm with different initial values but none of the run could generate any solution in this range of diameter. It can also be observed from the figures that the solutions obtained form OTTC algorithm are good in lower and higher range of diameter, however, the results obtained from RGH are good only in the lower range of the diameter. Contrary to this, MOEA is able to locate solutions in the higher range of the diameter with almost comparable quality of the solutions obtained by OTTC. The solutions obtained by OTTC in the middle range are much sub-optimal and are inferior to the solutions obtained by MOEA. In the upper-middle range of diameters, RGH could not locate solutions at all, and the solutions located in this range by OTTC are much inferior to the solutions obtained by MOEA. Thus, quality of the solutions obtained by MOEA is much superior in this range, and comparable in higher range to those of OTTC. To reflect this by rank-histogram, we include, for example, in Fig. 9 the rank-histogram of two solution-sets taken from MOEA and OTTC respectively; a long-tail reflects the inferior quality of solutions obtained from OTTC.

These are interesting observations, and are partly contrary to those reported by Raidl & Julstrom [9]. Raidl & Julstrom have shown that their technique works the best over all the other such techniques including OTTC. On looking from the plots in Figures 1, 5, 6 and 8, it can be observed that IR results are very close to RGH results in lower-diameter region, and little inferior in higher-diameter range to those of RGH. Moreover, IR could find competitive solutions in the entire range of the diameter which no other algorithm (barring MOEA) could do, is a significant achievement. However, since previous researchers could not visualize the solutions in the entire range of the diameter, their observations were biased. We reiterate that their conclusions were based on the experiments which they did for a particular value of the diameter and they could not evaluate the results over the entire range of diameter. In this work, since we could simultaneously obtain solutions for the entire range of the diameter, we could have a meaningful comparison of the existing algorithms too.

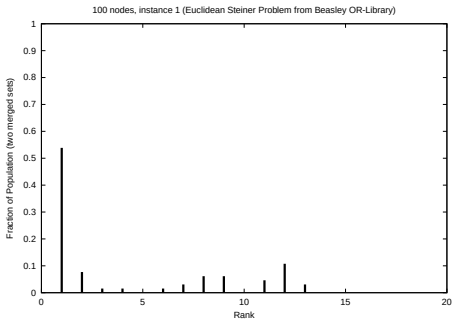

Fig. 9. Rank-histogram computed from OTTC and MOEA solution sets for 100 node data.

The above arguments are well supported by the metrics shown in Table 1. We are currently investigating the empirical behavior shown by these four algorithms, and how this knowledge can be used to further improve the solution-set by fine-tuning the evolutionary operators for the MOST problem.

5 Discussion & Conclusions

In this work, we demonstrated generating spanning trees subject to their satisfying the twin objectives of minimum cost and diameter. The obtained solution is a set of (near-optimal) spanning trees that are non-inferior with respect to each other. A network designer having a range of network cost and diameter in mind, can examine several optimal trees simultaneously and choose one based on these requirements and other engineering considerations.

To the best of our knowledge, this is the first work which attempts obtaining the complete Pareto front. Zhou & Gen [30] also obtained a set of solutions, they did not experiment on any benchmark data and, therefore, could not compare the quality of the solutions. It is shown by Knowles & Corne [12] that the front obtained by Zhou & Gen [30], was sub-optimal. We attribute the sub-optimality due to their use of an EA implementation which was unable to assess convergence. Knowles & Corne [12] used a weighted sum approach and could get comparable solutions but their approach is sensitive to the selection of weight values.

The work presented in this paper presents a generic framework which can be used to optimize any number of objectives simultaneously for spanning tree problems. The simultaneous optimization of objectives approach has merits over the constrained-based approaches, e.g., OTTC, IR and RGH algorithms. It is shown that the constrained-based approaches are unable to produce quality solutions over the entire range of the Pareto-front. For example, the best known algorithm of diameter-constrained spanning tree is RGH which is shown to be good for smaller values of diameters *only*, and is unable to produce solutions in the higher range. Similarly, the other well-known OTTC algorithm produces sub-optimal solutions in the middle range of the diameter. MOEA could obtain

superior solutions in the entire range of the objective-values. The solutions obtained by MOEA may further be improved marginally by proper tuning of evolutionary operators for the specific values of the objectives by introducing problem specific knowledge while designing evolutionary operators; such type of improvement, is however, difficult with an approximation algorithm.

Acknowledgements

Authors would like to thank anonymous reviewers for useful suggestions. A part of the research is supported from the Ministry of Human Resource Development (MHRD), Government of India project grant.

References

1. Garey, M.R., Johnson, D.S.: Computers and Interactability: A Guide to the Theory of NP-Completeness. San Francisco, LA: Freeman (1979)
2. Hochbaum, D.: Approximation Algorithms for NP-Hard Problems. Boston, MA: PWS (1997)
3. Marathe, M.V., Ravi, R., Sundaram, R., Ravi, S.S., Rosenkrantz, D.J., Hunt, H.B.: Bicriteria Network Design Problems. J. Algorithms **28** (1998) 142 – 171
4. Ravi, R., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Hunt, H.B.: Approximation Algorithms for Degree-Constrained Minimum-Cost Network Design Problems. Algorithmica **31** (2001) 58 – 78
5. Boldon, N., Deo, N., Kumar, N.: Minimum-Weight Degree-Constrained Spanning Tree Problem: Heuristics and Implementation on an SIMD Parallel Machine. Parallel Computing **22** (1996) 369 – 382
6. Deo, N., Kumar, N.: Constrained Spanning Tree Problems: Approximate Methods and Parallel Computations. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Math Society **40** (1998) 191 – 217
7. Deo, N., Abdalla, A.: Computing a Diameter-Constrained Minimum Spanning Tree in Parallel. In: Proc. 4th Italian Conference on Algorithms and Complexity (CIAC 2000), LNCS 1767. (2000) 17 – 31
8. Deo, N., Micikevicius, P.: Comparison of Prüfer-like Codes for Labeled Trees. In: Proc. 32nd South-Eastern Int. Conf. Combinatorics, Graph Theory and Computing. (2001)
9. Raidl, G.R., Julstrom, B.A.: Greedy Heuristics and an Evolutionary Algorithm for the Bounded-Diameter Minimum Spanning Tree Problem. In: Proc. 18th ACM Symposium on Applied Computing (SAC 2003). (2003) 747 – 752
10. Julstrom, B.A., Raidl, G.R.: Edge Sets: An Effective Evolutionary Coding of Spanning Trees. IEEE Trans. Evolutionary Computation **7** (2003) 225 – 239
11. Knowles, J.D., Corne, D.W.: A New Evolutionary Approach to the Degree-Constrained Minimum Spanning Tree Problem. IEEE Trans. Evolutionary Computation **4** (2000) 125 – 133
12. Knowles, J.D., Corne, D.W.: A Comparison of Encodings and Algorithms for Multiobjective Minimum Spanning Tree Problems. In: Proc. 2001 Congress on Evolutionary Computation (CEC-01). Volume 1. (2001) 544 – 551
13. Coello, C.A.C., Veldhuizen, D.A.V., Lamont, G.B.: Evolutionary Algorithms for Solving Multiojective Problems. Boston, MA: Kluwer (2002)

14. Deb, K.: Multiobjective Optimization Using Evolutionary Algorithms. Chichester, UK: Wiley (2001)
15. Coello, C.A.C., Lamont, G.B.: Applications of Multiojective Evolutionary Algorithms. World Scientific (2004)
16. Kumar, R., Rockett, P.I.: Improved Sampling of the Pareto-front in Multiobjective Genetic Optimization by Steady-State Evolution: A Pareto Converging Genetic Algorithm. Evolutionary Computation **10** (2002) 283 – 314
17. Kumar, R., Rockett, P.I.: Assessing the Convergence of Rank-based Multiobjective Genetic Algorithms. In: Proc. 2nd IEE/IEEE Int. Conf. Genetic Algorithms in Engineering Systems: Innovations and Applications (Galesia 97). Volume I446., IEE, London, UK (1997) 19 – 23
18. Kumar, R., Singh, P.K., Chakrabarti, P.P.: Multiobjective Genetic Search for Spanning Tree Problem. In: Proc. 11th Int. Conf. Neural Information Processing (IcoNIP), LNCS 3316, Kolkata, India, Springer-Verlag (2004) 218–223
19. Kumar, R., Singh, P.K., Chakrabarti, P.P.: Improved Quality of Solutions for Multiobjective Spanning Tree Problem Using EA. In: Proc. 11th Int. Conf. High Performance Computing (HiPC), LNCS 3296, Bangalore, India, Springer-Verlag (2004) 494 –503
20. Purshouse, R.C., Fleming, P.J.: Elitism, Sharing and Ranking Choices in Evolutionary Multicriterion Optimization. Research Report No. 815, Dept. Automatic Control & Systems Engineering, University of Sheffield (2002)
21. Deb, K.: A Fast Non-Dominated Sorting Genetic Algorithm for Multiobjective Optimization: NSGA-II. In: Proc. Parallel Problem Solving from Nature, PPSN-VI. (2000) 849 – 858
22. Zitzler, E., Laumanns, M., , Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In: Proc. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems (EUROGEN). (2001)
23. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining Convergence and Diversity in Evolutionary Multiobjective Optimization. Evolutionary Computation **10** (2002) 263 – 282
24. Hansen, M.P., Jaszkiewicz, A.: Evaluating the Quality of Approximations to the Nondominated Set. Tech. Rep. IMM-REP-1998-7, Tech. Univ. Denmark (1998)
25. Zitzler, E.: Evoluationary Algorithms for Multiobjective Optimization: Methods and Applications. Ph.D. Thesis, Swiss Federal Institute of Technology, Zurich, Switzerland (1999)
26. van Veldhuizen, D.A.: Multiobjective Evolutionary Algorithms: Classifications, Analysis, and New Innovations. Ph.D. Thesis, Technical Report No. AFIT/DS/ENG/99-01, Air Force Institute of Technology, Dayton, OH (1999)
27. Schott, J.R.: Fault Tolerant Design Using Single and Multicriteria Genetic Algorithms. Masters Thesis, Department of Aeronautics and Astronautics, MIT, Massachusetts (1995)
28. Deb, K., Jain, S.: Running Performance Metrics for Evolutionary Multiobjective Optimization. In: Proc. 4th Asia-Pacific Conf. Simulated Evoluation and Learning (SEAL 02), Singapore (2002) 13–20
29. Knowles, J., Corne, D.: On Metrics for Comparing Nondominated Sets. In: Proc. Congress Evolutionary Computation (CEC '02). Volume I., Piscataway, NJ, IEEE (2002) 711–716
30. Zohu, G., Gen, M.: Genetic Algorithm Approach on Multi-Criteria Minimum Spanning Tree Problem. European J. Operations Research **114** (1999) 141–152
31. Soak, S.M., Corne, D., Ahn, B.H.: A powerful new encoding for tree-based combinatorial optimisation problems. In: Parallel Problem Solving from Nature - PPSN VIII. Volume 3242 of LNCS., Birmingham, UK, Springer-Verlag (2004) 430–439
32. Cantu-Paz, E.: Efficient and Accurate Parallel Genetic Algorithms. Boston, MA: Kluwer (2000)