Energy-efficient Multicast Routing by using Genetic Local Search

Valeri Katerinchuk[1], Andreas Albrecht[2] and Kathleen Steinhöfel[1]
[1]Department of Informatics, King's College London, Strand Campus, London, U.K.
[2]School of Science and Technology, Middlesex University London, The Burroughs, London, U.K.

Keywords: Energy Efficiency, Wired Networks, Multiple Multicast Requests, Multicast Routing, Simulated Annealing, Genetic Algorithm, Hybrid Algorithm, Partially Mixed Crossover.

Abstract: Energy-efficient multicast routing algorithms have predominantly focused on wireless or ad-hoc mobile networks. However, since the turn of the century the need for energy efficient approaches to routing in wired networks has been steadily rising. In this paper, we introduce an objective function for multicast routing in wired networks taking energy consumption into consideration. A number of hybrid Genetic and Simulated Annealing based algorithms have been shown to be able to find better solutions to the multicast routing problem compared to solely Genetic or Simulated Annealing based algorithms. Our approach adapts a population-based hybrid algorithm for routing multiple simultaneous multicast requests. We examine the performance in terms of energy efficiency against solutions found by Logarithmic Simulated Annealing and Genetic based algorithms. We find that the hybrid approach, in 87% of instances, was able to find superior solutions, and in 96% of instances, solutions superior or equal to the best solution given by either Simulated Annealing or Genetic approaches. The extent of the improvement however varied greatly from a few hundred to within ten Joules, with the improvement on the best solution ranging from 5.6 to 531.5 Joules.

1 INTRODUCTION

Multicasting is the transmission of data simultaneously from one source to multiple destinations within a telecommunications network. Unlike the more inefficient point-to-point methodology of Unicasting, Multicasting takes advantage of parallelism in the network. The Multicast Routing Problem (MRP) is the problem of routing one or many such multicast requests through a network, often by means of constructing a spanning tree, while minimising the usage of one or more limited network resources.

Since the MRP had been shown to be NP-complete in (Karp, 1972), numerous heuristics have been proposed to find near-optimal solutions. In (Wang et al., 2003) a Simulated Annealing (SA) based algorithm for the MRP is presented which was shown by tests against the benchmark National Science Foundation network to find solutions within 5% of the optimum. Genetic algorithms (GA) have been shown to obtain comparable solutions (Zhang and Leung, 1999), while demonstrating a greater variance in solution quality, obtained results closer than 2% to the optimal in an average of 91% of runs. Work by (Wang et al., 2006) on three implemented algorithms has also shown GA and a SA based approaches to be comparable, with the GA able to obtain slightly superior solutions while the SA based algorithm obtained solutions faster and performed better on inputs the GA struggled to obtain near-optimal solutions on. Recent work examining hybrid GA and SA based algorithms has shown them to generally be able to locate superior solutions to purely GA or SA approaches when applied to routing both a single (Xu et al., 2013) and multiple multicast requests (Zahrani et al., 2008).

Predominantly, papers considering algorithms for multicast routing define the MRP as a single request routing problem, where a message must be routed to all members of a single multicast group. However, realistically networks utilising protocols capable of true multicast routing are commonly faced with a number of simultaneous requests over multiple multicast groups. Unfortunately, the number of algorithms dealing with optimisation for multiple simultaneously routed multicast requests is few and primarily focused on Quality of Service (QoS) (Galiasso and Wainwright, 2001; Zahrani et al., 2008).

Moreover, the majority of algorithms for the MRP have featured either a QoS based cost function or a generic cost metric across each edge of a multicast routing (Pinto and Barn, 2006; Xu, 2011). A growing number of papers are considering the importance of

Katerinchuk V., Albrecht A. and Steinhöfel K..
Energy-efficient Multicast Routing by using Genetic Local Search.
DOI: 10.5220/0004926907400746
In *Proceedings of the 6th International Conference on Agents and Artificial Intelligence* (ICAART-2014), pages 740-746
ISBN: 978-989-758-015-4

energy conservation in wireless and ad-hoc networks (Banerjee et al., 2003; Olagbegi and Meghanathan, 2010; Xi and Yeh, 2010). For wired networks the need for energy-efficient algorithms is steadily growing. The last decade has seen the number of internet users rising rapidly with only 8% of the worlds population in 2001 to 38.8% in 2013 according to the Global ICT developments statistics provided by the Internet Telecommunications Union. By contrast, routing and wired connectivity equipment are constrained by technology limitations. As the data traffic increases, so does the energy required for the transmission of data and the need for energy-efficient routing. Another facet of this increase in energy consumption is the contribution network routing makes to rising CO_2 levels. The SMART 2020 study suggests the internet could account for 12% of global CO_2 levels by 2020. Existing papers considering energy efficient multicast routing rarely maintain an energy function considering energy loss factors unique to wired networks. In his paper (Ajibesin et al., 2013) introduces an energy function focusing primarily on the amount of data per unit of time as the measure of energy usage. A number of other papers(Lun et al., 2006; Xu and Qu, 2009) a general cost function over the links of the routing. However, there exist multiple factors influencing energy expenditure over wired networks.

In our paper we present an adaptation of the algorithm presented in (Zahrani et al., 2008) toward energy-efficient routing for the MRP. We propose a new energy function for the capacity constrained MRP representing the energy cost of multicast routing in wired telecommunications networks.

We test the implemented algorithm over a number of network instances taken from the Beasley JE. OR library, calculating the energy cost of the edges using the proposed energy function, against a Logarithmic Simulated Annealing and Genetic algorithms. We find that in the majority of cases the hybrid approach is able to find solutions closer to or as near the optimum as the best solutions found by the other two algorithms.

2 FORMALISATION

Communication networks consist of nodes connected through links. The nodes transmit and receive information, while the links transport information between the nodes. Links have a limited capacity for carrying information at any one time.

A communication network is represented by graph $G = (V,E)$ where node set $V = \{m_1, m_2..m_i\}$ nodes and a set of links $E = \{l_1, l_2..l_i\}$ where each link l_j connects two nodes in V. A link $l = (v_i, v_j, c)$ where $v_i, v_j \in V$, c is a capacity limit on the information that can traverse at one time over l minus the capacity requirement of the current traffic over l. We also define an energy cost function Co over E such that each link l_i in E has an associated cost of transmitting data over that link $Co(l_i)$. A multicast routing request is defined as

$$m = [s \Rightarrow D, C],$$

where $s \in V$ is the source node of m, $D = \{v_1, v_2, ..v_i\} \subseteq V$ is the set of destination nodes and C is the capacity required by each transmission $s \Rightarrow v_i$ and represents bandwidth lost due to the amount of data to be transmitted and the routing protocol used by the message.

The MRP for multiple simultaneous multicast routing requests is defined as

$$[P = G; Co; M],$$

where $M = \{m_1, m_2, ..m_i\}$. We adopt a Steiner tree representation of the routing of a single multicast routing request over G. All data is routed over G simultaneously.

2.1 Energy Cost Function

A successful routing of request $m_i \in M$ is a subgraph $G_r = (V_r, E_r)$ representing a Steiner tree over G, where for every link $l \in E_r$, $c(l) \geq 0$. There are numerous factors which influence the amount of energy necessary for a routing separated broadly into energy loss in the nodes, energy loss in the links and initial cost of transmission, dependent on data volume. The energy required E_t for a multicast routing is therefore defined as

$$E_t = E_{in} + \sum_{V_r}(E_v) + \sum_{E_r}(E_l),$$

where E_{in} is the initial energy required to transmit the message, E_{v_i} is the energy loss at vertex i and E_{l_i} is the energy lost during transmission over link i.

The initial energy required to transmit a message is taken as a function of the amount of data being transmitted. It is defined as $E_{bit} * n_{bit}$, where E_{bit} is the energy per bit, taken as 0.09/8=0.01125 joules - the maximum estimated energy cost of sending a bit of data over the internet (Gupta and Singh, 2003). By n_{bit} we denote the number of bits of data to be multicast over the network, which is set to 1000 bits per request rendering $E_{in} = 11.25$ joules.

The energy loss, $E_{v_i} = k$, associated with transmitting a request over a router i, in active mode, is a constant for each node given by

$$k = L_d + L_m + 2log_2 N(L_s), \tag{1}$$

where L_d and L_m are multiplexer and demultiplexer losses respectively, L_s is the switch element insertion loss and N is the number of input/output ports of the switch with the relation $2log_2N(L_s)$ representing the total energy lost in the cable-independent matrix switch (Ramamurthy et al., 1999).

The energy lost routing the request over $l_i \in E_r$, $E_{l_i} = E_{in}(l_i) - E_{out}(l_i)$ with $E_{in}(l_i)$ being the energy of the transmission entering the link and $E_{out}(l_i)$ the energy of the transmission exiting the link. As the signal is amplified to compensate for any energy lost when exiting a node we may assume $E_{in}(l_i) = E_{in}$. The energy loss in l_i is dependant on the attenuation α of the wire over its length. The attenuation of the wire is given in (Rajagopal, 2007; Kuphaldt, 2013) as

$$\alpha = 10log_{10}\frac{E_{in}(l_i)}{E_{out}(l_i)}, \tag{2}$$

in decibels per unit length, and varies depending on the type of wire. For the purpose of this paper we assume all links are facilitated by the popular category 5e (cat5e) wires. The TIA-EIA-568 standards document gives the formula for calculating the attenuation wire based on frequency and wire type used f as

$$\alpha = k_1\sqrt{f} + k_2 * f + \frac{k_3}{\sqrt{f}}\ db/100m, \tag{3}$$

where $k_1 = 1.967, k_2 = 0.023$ and $k_3 = 0.05$ are constants specific to the wire type. Frequency f is taken as 100Mhz - the highest allowable frequency for the cat5e wire given by the TIA-EIA sandards. Therefore the attenuation per 100m of cat5e wire is $\alpha = 21.975$db (Joules per 100m per second). The output power of a link $E_{out}(l_i)$ can then be found by

$$E_{out}(l_i) = E_{in}(l_i)/10^{\frac{\alpha*(ln/100)}{10}}, \tag{4}$$

where ln is the length of the wire in meters.

However, as the output power drops rapidly with distance it must be amplified regularly to prevent the signal becoming too weak to identify. The TIA-EIA standards recommend a maximum length of cable of 100m in order for the signal to remain recognisable, and we will assume that every link will have an amplifier at 100m intervals which boosts the signal to starting levels. The total energy loss over link l_i then becomes

$$E_l = (E_{in}(l_i) - E_{out}(l_i)) * a_n + E_{l_c}, \tag{5}$$

where a_n is the number of amplifiers, $E_{out}(l_i)$ now represents energy after 100m of wire and $E_{l_c} = E_{in}(l_i) - E_{in}(l_i)/10^{\frac{\alpha*(l_c/100)}{10}}$, is the energy lost due to attenuation over the length of wire $l_c \leq 100$, not compensated for by amplifiers.

Assuming as before that $E_{in}(l_i) = E_{in} = 0.01125 * n_{bit}$, and that n_{bit} is constant for all transmissions (1000 for the purpose of this study), the energy lost after 100m can now be represented as a constant

$$E_{in}(l_i) - E_{in}(l_i)/10^{\frac{\alpha*(100/100)}{10}} = 11.25, \tag{6}$$

measured in Joules per second. The energy lost transmitting along edge l_i can be summarised as

$$E_l = 11.25 * a_n + E_{l_c}, \tag{7}$$

As every link l in a Steiner tree representing a successful routing G_r of m_i is connected to exactly one originator v_i and destination node v_j, k may be safely added to the total energy loss of transmitting over link l_i

$$E_t = E_{in} + \sum_{E_r}(11.25 * a_n + E_{l_c} + k), \tag{8}$$

rendering it irrelevant for the purpose of minimising the energy function. Similarly, $E_{in} = 11.25$ in the above equation represents a constant for every multicast routing request and can be ignored for the purpose of finding the minimum cost routing resulting in the final equation representing energy cost of a routing over G as

$$E_t = \sum_{E_r}((E_{in}(l_i) - E_{in}(l_i)/10^{\frac{\alpha*(100/100)}{10}}) * a_n + E_{l_c}), \tag{9}$$

which may further be reduced, given (6), to

$$E_t = \sum_{E_r}(11.25 * a_n + E_{l_c}). \tag{10}$$

with

$$Co(l_i) = 11.25 * a_n + E_{l_c} \tag{11}$$

3 HYBRID MULTICAST ROUTING ALGORITHM

In this paper we adopt the genetic local search with pre-processing by logarithmic simulated annealing algorithm presented in (Zahrani et al., 2008). The problem of minimising $\sum_E Co(l_i)$ over G, subject to s, D and C has been shown to be NP-complete (Karp, 1972). As each request may need to be rerouted repeatedly in order to satisfy capacity constraints and as part of the SA and elitist Partially Mixed Crossover (PMX) approaches it employs the KMB(Kou et al., 1981) heuristic algorithm to estimate the Steiner tree for each network routing. The KMB has been shown, on average, to produce Steiner trees 5% over the cost of a minimum Steiner tree(Doar and Leslie, 1993) and has been known to find minimum values for numerous benchmark problems (Koch and Martin, 1998).

The neighbourhood operator used by both the SA and GA stages of the algorithm takes a pair of random numbers between 1 and r, where r is the total number of multicast requests. The order of all requests between these two numbers is reversed. This is done in order to preserve any desirable dependencies of requests resulting in low cost solutions. The algorithm utilises a two stage process to estimate an optimal routing of multiple multicast routing requests:

3.1 LSA Pre-processing

The primary purpose of the LSA pre-processing step is to perform an energy landscape analysis refining an estimate of Γ, representing the maximum escape height from local minima in the solution space. That is to say, an estimate of the difference in cost of a worse accepted solution when compared to a solution representing a local minima that must be met in order to guarantee exiting any local minima. Initial solutions for the algorithm are generated by the KMB and an initial estimate of Γ is obtained by

$$\Gamma = \frac{G_{est}}{10}, \tag{12}$$

with G_{est} representing the difference between the best and worst solutions found, initially set to the difference between two initial solutions. The algorithm derives initial temperature from the estimate of Γ and maintains a logarithmic cooling schedule. Throughout its execution the Simulated Annealing algorithm constantly updates G_{est} with the difference between the current best and worst solution. Upon the termination of the LSA gamma is re-estimated using (12) updated with the new value for G_{est}. The best solution found by the LSA is added to the initial population of the Genetic PMX algorithm. As the Genetic algorithm is elitist, always conserving the current best solution between runs, this ensures that good solutions found by the LSA are conserved.

```
Begin
    Generate initial solution;
    Generate initial value for G_est;
    Generate initial value for Gamma;
    Calculate initial temperature;
    for (int i=0; i<constant; i++){

       Generate neighbouring solution N;
       if (cost(N)>currentworst){
          currentworst+= cost(N);
       }
       if (cost(N)<cost(Bestsolution){
           Bestsolution = N;
           G_est=currentworst-cost(Bestsolution)
       }
       if (cost(current)>cost(N)){
           accept neighbour solution;
      }
      else{
           accept neighbour solution
           with probability determined
           by current temperature;
       }
   }
End.
```

3.2 Genetic Local Search

Upon initialisation, a population of initial solutions is generated using the KMB. The bulk of the Genetic Local search algorithms consists of three iterated steps. For each solution in the population, initially the algorithm performs only downward steps, accepting only solutions which improve upon the cost. Once a local minima is reached the algorithm switches modes performing upward steps until the escape height $h = \Gamma_{est} + C_o(E_r(m_i))$ is reached where Γ_{est} is the estimate of gamma provided by the pre-processing step. The above procedure is repeated for every solution in the population. The algorithm then performs a PMX operation, adding, in the first run the best solution generated by the Simulated Annealing step. The three steps are repeated until a set number of PMX operations have occurred. The best current solution is conserved between PMX operations.

```
Begin
    Generate initial solutions;
    Obtain Gamma_est from LSA;
    for (N steps){
       for (each solution in population){
           for (M steps){
                While (true){
                     Begin downward steps;
                     if (local minima is reached){
```

```
                        break;
                    }
                }
                While (true){
                    Begin upward steps;
                    if (Gamma_est+cost(minima)
                    is reached){
                        break;
                    }
                }
            }
            Begin PMX;
            save current best solution;
        }
    }
End.
```

4 COMPUTATIONAL EXPERIMENTS

We have implemented the algorithm described in section (3). For the computational experiments the benchmark problem instances of steinb6 (50 nodes, 100 edges), steinb7 (75 nodes, 94 edges), steinb8 (75 nodes, 94 edges), steinb10 (75 nodes, 150 edges), steinb11 (75 nodes, 150 edges) and steinb18 (100 nodes, 200 edges) were taken from the Beasley JE. OR library to represent the input graphs G. Edge costs were extrapolated from the problem instances by applying the energy function to the edge costs given in the steinb instances where the steinb edge costs were taken to represent distance of the links in hundreds of meters. As detailed in (2.1), we have taken the energy/bit from (Gupta and Singh, 2003) to be 0.001125 joules and have set the number of bits per transmission to 1000. The capacity limit c of each edge is initially set to 12.

For the energy function, we assume no loss due to deformation or imperfections in the wire. Amplifiers are assumed to be positioned along the links at 100m intervals as specified by the TIA-EIA standards document. The energy lost transmitting 1000 bits of data over 100m is rounded down to 11.2 Joules/second. Once a routing of all multicast requests is established, all transmissions are considered instantaneous (taken as 1 second for the purposes of the experiment). We assume all nodes in the graphs to be identical T640 routers in active mode.

The steinb 6, 7 and 8 instances were tested against a single request set of 9 multicast requests given in Table 1. The algorithm was run five times for each input instance.

For the steinb10,11 and 18 instances twenty multicast routing requests were obtained from (Zahrani et al., 2008) shown in Table 2.

Table 1: Set of multicast requests for steinb 6, 7 and 8.

m_i	s	D	C
1	36	7,23,25,40	3
2	17	15,30,31,40,41,46	2
3	48	36,50	5
4	2	6,14,18,23,27,33,47,49	4
5	41	13,22,27,35,50	2
6	30	5,12,28,31,44,45	2
7	23	13,14,28,41 ,35,45	1
8	10	5,20,31,40	3
9	16	18,20,22,23	2

Table 2: Set of multicast requests for steinb11,10 and 18.

m_i	s	D	C
1	24	20.29.30	3
2	55	4,21,41	5
3	10	5,20,31,40	3
4	41	13,22,27,35,50	2
5	17	15,30,31,40,41,46	2
6	14	6,16,36,4	4
7	67	23,29	6
8	69	10,40,54	2
9	13	28	7
10	53	13,14,28,41,52,55	1
11	50	5,12,28,31,44,45	2
12	36	7,23,25,40	3
13	52	9,13,22,55	2
14	2	6,14,18,23,27,33,47,49	4
15	48	36,58	8
16	61	15,20,33,38	6
17	14	9,16,31,43,44	3
18	9	4,6,7,30,31,35	2
19	66	18,20,22,23	2
20	75	33,57	7

These were formed into three sets of multicast requests M_1, M_2 and M_3 containing 9, 15 and 20 multicast requests respectively. For M_1 and M_2 the algorithm was run five times on two of the three input instances and five times on all three input instances for M_3.

In the experiments we compared the quality of solutions found by the joint PMX and LSA algorithms to those obtained by running the LSA and PMX with random walk alone. Tables 3-5 show the total energy values for solutions obtained when routing multicast requests across steinb 6,7 and 8 respectively, when routed by LSA only, PMX only and PMX with LSA pre-processing algorithms. Tables 6-8 show the total energy values for solutions obtained when routing multicast requests sets M_1, M_2 and M_3 across steinb10,11 and 18 respectively, when routed by LSA only, PMX only and PMX with LSA pre-processing algorithms.

Table 3: Total energy cost of steinb6 solutions.

P_i size	E_t LSA	E_t GA	E_t GA with LSA
9	3855.6	3785.1	3785.1
9	3691.4	3844.3	3622.3
9	3785.1	3817.9	3757.6
9	3743.8	3759.8	3690.1
9	3706.5	3747.2	3697.5

Table 4: Total energy cost of steinb7 solutions.

P_i size	E_t LSA	E_t GA	E_t GA with LSA
9	7613.1	7682.1	7447.8
9	7672.0	7541.2	7380.8
9	7593.6	7832.3	7511.1
9	7672.0	7671.2	7394.8
9	7627.8	7911.0	7434.6

Table 5: Total energy cost of steinb8 solutions.

P_i size	E_t LSA	E_t GA	E_t GA with LSA
9	5635.7	5789.1	5600.4
9	5644.2	5698.5	5571.3
9	5670.8	5666.3	5571.3
9	5611.2	5691.5	5667.2
9	5743.9	7604.3	5639.0

Table 6: Total energy cost of steinb10 solutions.

P_i size	E_t LSA	E_t GA	E_t GA with LSA
9	3940.9	4126.3	3789.5
9	3957.0	3899.1	3832.7
9	3835.7	4011.2	3701.5
9	3921.2	4046.6	3835.0
9	3947.0	4021.6	3947.0
20	9497.6	9532.9	9094.4
20	9534.2	9965.1	9295.3
20	9508.8	9678.3	9150.9
20	9464.0	9772.9	9229.1
20	9564.8	9893.3	9216.6

We found that in 96% of instances the solution found by the hybrid algorithm was superior or equal to the best solution found by either the LSA or the PMX algorithms alone. Interestingly, in a large number of instances the LSA outperformed the PMX with random walk and was able to find a superior solution.

5 CONCLUSIONS

In this paper we presented a new energy function for the energy-efficient Multicast Routing Problem

Table 7: Total energy cost of steinb11 solutions.

P_i size	E_t LSA	E_t GA	E_t GA with LSA
15	4233.6	4413.2	4233.6
15	4222.4	4402.2	4135.9
15	6619.2	4423.5	4217.0
15	4222.4	4472.9	4188.0
15	4233.6	4426.1	4153.4
20	5495.2	5331.9	5226.8
20	5693.6	6413.3	5471.8
20	5555.9	5934.1	5336.4
20	5667.2	6117.5	5451.5
20	5492.4	5423.3	5417.7

Table 8: Total energy cost of steinb18 solutions.

P_i size	E_t LSA	E_t GA	E_t GA with LSA
9	3483.2	3532.5	3389.3
9	3640.0	3789.3	3570.4
9	3572.8	3711.4	3492.0
9	3495.5	3752.4	3411.6
9	3547.1	3669.5	3455.6
20	8681.1	8712.8	8681.1
20	8642.0	9030.6	8598.8
20	8615.1	8761.7	8083.6
20	8742.8	9136.2	8766.8
20	8791.3	8881.9	8778.0

in wired networks. We extended the algorithm presented in (Zahrani et al., 2008) to calculate energy efficient solutions to the MRP over multiple simultaneously routed multicast requests. We have shown that an energy-efficient PMX with random walk algorithm employing LSA pre-processing was able to find, 87% of the time, superior solutions than either LSA or PMX alone, 96% of the time, solutions superior or equal to the best solution given by LSA or PMX. The extent of the improvement however varied greatly from a few hundred Joules to within twenty. The range of improvement on the best solution for steinb 6, 7 and 8 instances was from 9 to 276.4 Joules. The range of improvement on the best solution for steinb 10, 11 and 18 instances was between 5.6 and 531.5 Joules. Future work may include considering the impact of routing requests with varying amount of data on the routing of multiple multicast request sets and incorporating into the energy function energy loss due to the protocol used.

REFERENCES

Ajibesin, A. A., Wajiga, G. M., Odekunle, M. R., and Egunsola., O. K. (2013). Energy-efficient multicast in wired and wireless networks: Analysis and performance measures. In *Computational Intelligence, Communication Systems and Networks (CICSyN), 2013 Fifth International Conference on*, pages 131–136.

Banerjee, S., Misra, A., Yeo, J., and Agrawala, A. (2003). *Energy-Efficient Broadcast and Multicast Trees for Reliable Wireless Communication*.

Doar, M. and Leslie, I. M. (1993). How bad is naive multicast routing. In *IEEE INFOCOM.*, volume 1, pages 82–89.

Galiasso, P. and Wainwright, R. L. (2001). A hybrid genetic algorithm for the point to multipoint routing problem with single split paths. In *Proceedings of the 2001 ACM Symposium on Applied Computing*, SAC '01, pages 327–332. ACM.

Gupta, M. and Singh, S. (2003). Greening of the internet. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, pages 19–26. ACM.

Karp, R. M. (1972). Reducibility among combinatorial problems. In Miller, R. E., Thatcher, J. W., and Bohlinger, J. D., editors, *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Springer US.

Koch, T. and Martin, A. (1998). Solving steiner tree problems in graphs to optimality. *Networks*, 32:207–232.

Kou, L., Markowsky, G., and Berman, L. (1981). A fast algorithm for steiner trees. *Acta Informatica*, 15:141–145.

Kuphaldt, T. R. (2000-2013). *Lessons In Electric Circuits – Volume 3*.

Lun, D. S., Ratnakar, N., Mdard, M., Koetter, R., Karger, D. R., Ho, T., Ahmed, E., and Zhao, F. (2006). Minimum-cost multicast over coded packet networks. *IEEE TRANS. INF. ON THE*, pages 2608–2623.

Olagbegi, B. S. and Meghanathan, N. (2010). A review of the energy efficient and secure multicast routing protocols for mobile ad hoc networks. *International Journal on Applications of Graph Theory In wireless Ad Hoc Networks And sensor Networks*, 2:1–15. arXiv:1006.3366 [cs].

Pinto, D. and Barn, B. (2006). Multiobjective multicast routing with ant colony optimization. In *Network Control and Engineering for Qos, Security and Mobility, V*, pages 101–115.

Rajagopal, K. (2007). *Engineering Physics*. PHI Learning Pvt. Ltd.

Ramamurthy, B., Datta, D., Feng, H., Heritage, J. P., and Mukherjee, B. (1999). Impact of transmission impairments on the teletraffic performance of wavelength-routed optical networks. *Journal of Lightwave Technology*, 17:1713–1723.

Wang, X., Cao, J., Cheng, H., and Huang, M. (2006). QoS multicast routing for multimedia group communications using intelligent computational methods. *Comput. Commun.*, 29:2217–2229.

Wang, X., Cheng, H., Cao, J., Zheng, L., and Huang, M. (2003). A simulated-annealing-based QoS multicasting algorithm. In *International Conference on Communication Technology Proceedings, 2003. ICCT 2003*, volume 1, pages 469–473 vol.1.

Xi, Y. and Yeh, E. (2010). Distributed algorithms for minimum cost multicast with network coding. *IEEE/ACM Transactions on Networking*, 18:379–392.

Xu, Y. (2011). *Metaheuristic Approaches for QoS Multicast Routing Problems*. PhD thesis.

Xu, Y. and Qu, R. (2009). A grasp approach for the delay-constrained multicast routing problem. In *Proceedings of the 4th Multidisciplinary International Conference on Scheduling: Theory and Applications*, pages 99–104.

Xu, Y., Qu, R., and Li, R. (2013). A simulated annealing based genetic local search algorithm for multi-objective multicast routing problems. *Annals of Operations Research*, 206:527–555.

Zahrani, M. S., Loomes, M. J., Malcolm, J. A., Ullah, A. Z. M. D., Steinhoefel, K., and Albrecht, A. A. (2008). Genetic local search for multicast routing with pre-processing by logarithmic simulated annealing. *Computers & Operations Research*, 35:2049–2070.

Zhang, Q. and Leung, Y.-W. (1999). An orthogonal genetic algorithm for multimedia multicast routing. *Transactions on Evolutionary Computation*, 3:53–62.