Finding Optimal Paths in MREP Routing

Rudolf Fleischer[1] Mordecai Golin[1] Chin-Tau Lea[2*] Steven Wong[2]

[1] Dept of CS, HKUST, Clear Water Bay, Hong Kong, `[rudolf,golin]@cs.ust.hk`
[2] Dept of EEE, HKUST, Clear Water Bay, Hong Kong, `eelea@ee.ust.hk`

Keywords: approximation algorithm, graph algorithms, computational complexity

Abstract

Maximum Residual Energy Path (MREP) routing has been shown an effective routing scheme for energy conservation in battery powered wireless networks. Past studies on MREP routing are based on the assumption that the transmitting node consumes power, but the receiving node does not. This assumption is false if acknowledgement is required as occurs, for example, in some Bluetooth applications.

If the receiving node does not consume power then the MREP routing problem for a single message is easily solvable in polynomial time using a simple Dijkstra-like algorithm. We further show in that when the receiving node does consume power the problem becomes NP-complete and is even impossible to approximate with an exponential approximation factor in polynomial time unless $P = NP$.

1 Introduction

Recent advances in wireless technologies, such as Bluetooth [10], have made it easy and practical to construct an ad hoc network for novel applications — a surveillance network, a wireless tag network in a grocery store, and a sensor network to monitor environment dangerous conditions are just a few examples.

Battery power is a precious resource for each wireless device (except the wired gateway computer) and energy conservation is critical for such a network. Routing will play an important role in energy conservation. This issue has been studied extensively in the past [2, 3, 9, 11, 14, 15, 16]. A central part of any routing study is the definition of the path metric. Some metrics combine both delay and power consumption, whereas others focus on maximizing the system life time and ignore the delay.

When delay is less a concern than system life, *Maximum Residual Energy Path (MREP)* routing has been shown an effective scheme for energy conservation [2, 3, 4]. In MREP routing, the best path is one that maximizes the energy of that node on the path with the least energy after sending the message. Previous work on MREP routing concentrated on heuristics for a constant stream of messages to be routed through the network [2, 3, 4], the case of hybrid cost functions that try to balance total energy consumption and energy drain at a single node [12, 15], or just studied local routing heuristics in a distributed system [17]. All these studies have been based on the assumption that sending packets requires energy, but receiving packets does not. In this case, we show that the

*Research is supported by the Hong Kong RGC grant HKUST6201/03E

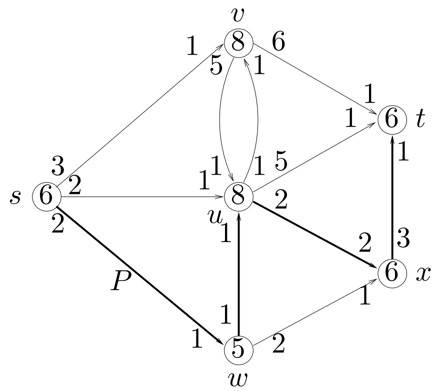


Figure 1: A routing network. Nodes are labeled by their energy. Outgoing edges are labeled by the send cost along the edge, and incoming edges are labeled by the acknowledgement cost. The path $P = (s, w, u, x, t)$ has $D(P) = 1$ (with node x being the bottleneck) and is therefore legal.

MREP routing problem for a single message is reducible to the max-bandwidth path problem and therefore polynomial-time solvable using a standard Dijkstra-type algorithm (a similar algorithm is sketched in [2]).

However, the assumption of energy-free reception is not always justified. Ad hoc networking technologies normally include energy-conservation states. Between two transmissions, nodes in the network may enter the energy-conservation state (sleeping) state. A sleeping node along the chosen path has to be waken up first. The process requires a hand-shaking packet exchanged between the sender and the receiver. Sending a hand-shaking packet by the receiver will cost energy. Another example is if the ad hoc network has very noisy connections (like an earthquake monitoring sensor network or a battlefield network), hop-by-hop acknowledgement may be used. For every packet received, the receiver must send back an acknowledgement packet. This costs energy. We show that in this case the MREP routing problem for a single message is NP-complete and is impossible even to approximate with an exponential approximation factor in polynomial time unless $P = NP$.

After defining the MREP routing problem in Section 2, we will show in Section 3 that it can be efficiently solved in networks without acknowledgement costs by a simple transformation to the max-bandwidth path problem. We then show in Section 4 that the general problem with acknowledgement costs is NP-complete and very hard to approximate.

2 Definitions

We can model the MRE routing problem as follows. Let $G = (V, E)$ be a directed graph where V is a set of vertices (wireless nodes) and E is a set of directed edges (connections). Initially, each vertex u has a battery charged with *energy* $E_u \geq 0$. This energy will be consumed by sending or acknowledging messages. We usually just speak of the energy of a node.

With each edge $e = (u, v)$ we associate two costs, the *sending cost* s_e (or $s_{u,v}$) for sending a message along e, i.e., the power used up by the sender when transmitting the full message, and the *acknowledgement cost* r_e (or $r_{u,v}$) for receiving a message, i.e., the power used by the receiver to transmit an acknowledgement message back to the sender. Sending a message along $e = (u, v)$ will reduce E_u, the energy of u, by s_e, and it will reduce E_v by r_e. Of course, sending a message is only possible if neither node energies become negative. Normally, s_e and r_e will depend on the distance between the vertices and on the size of the message (in our definition we assume all messages have unit size), but that is not important for our results.

When sending a message from a vertex s to a vertex t in G we can usually choose between many different paths along which we could route the message. We are interested in paths that leave a high energy in all the nodes on the path, i.e., we try to avoid situations where routing a message would use up all the energy of one node. Otherwise, we might have trouble routing the next message (the network could even become disconnected).

Formally, if we route a message along a path $P = (v_1, v_2, \ldots, v_{k-1}, v_k)$ in G, where $v_1, \ldots, v_k$ are vertices and $(v_1, v_2), \ldots, (v_{k-1}, v_k)$ are edges, then each node on P loses some energy due to the send and acknowledgement costs. Let R_u denote the *residual energy* of u after routing the message. If P is simple then $R_{v_1} = E_{v_1} - s_{v_1,v_2}$, $R_{v_i} = E_{v_i} - r_{v_{i-1},v_i} - s_{v_i,v_{i+1}}$ for $i = 2, \ldots, k-1$, and $R_{v_k} = E_{v_k} - r_{v_{k-1},v_k}$. Otherwise, P contains some vertex (or vertices) several times, and then we have to add up all the send and acknowledgement costs for that vertex. We now define the *minimum residual energy (MRE)* of P, denoted by $D(P)$, to be

$$D(P) = \min_{i=1,\ldots,k} \{R_{v_i}\} .$$

Note that we can only send a message along P if all nodes on P have non-negative residual energy, i.e., $D(P) \geq 0$. We call such a path *legal*. See Fig. 1 for an example.

We call a path *optimal* if it has maximum MRE among all paths that route a message from a given start vertex to a given end vertex. Routing along a nonsimple path is obviously never a good idea, so optimal paths are always simple. Our goal is to find optimal $s-t$ paths.

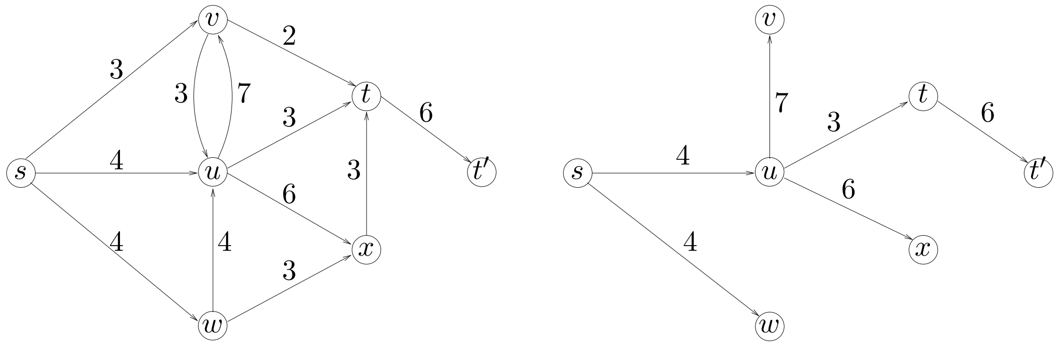
> MREPP **(Maximum Residual Energy Path Problem)**
>
> *Input:* A directed graph $G = (V, E)$ with vertex energies E_u, and send costs $s_{u,v}$ and acknowledgement costs $r_{u,v}$ on the edges (u, v). There are also two distinguished vertices s (source) and t (destination).
>
> *Output:* An $s-t$ path in G that maximizes $D(P)$ among all $s-t$ paths P in G.

3 MREPP **Without Acknowledgement Costs**

3.1 The Max-Bandwidth Problem

In this section we quickly review the max-bandwidth problem for graphs with bandwidth constraints on the edges and its solution. We will present an algorithm that will be used in the next section as a subroutine to solve MREPP without acknowledgement costs.

Let $G = (V, E)$ be a directed graph such that associated with each edge $(u, v) \in E$ there is a *bandwidth* $b(u, v) \geq 0$. Let $P = (v_1, v_2, \ldots, v_{k-1}, v_k)$ be a path in G. The *bandwidth* of P is $B(P) = \min_{1 \leq i < k} b(v_i, v_{i+1})$, i.e., the smallest bandwidth among all edges on P (also called the *bottleneck* of P). Given $s, t \in V$, the *max-bandwidth problem* is to find a path from s to t that maximizes $B(P)$. This problem has been well studied and it is well known that such a path can be found using a simple modification of Dijkstra's shortest path algorithm [5, 6]. A good history and description of the modified algorithm using modern notation is available in [13]. The standard implementation of this algorithm runs in $O(|E| \log |V|)$ time; with more sophisticated data structures this can be further reduced to $O(|E| + |V| \log |V|)$ [7].



(a) (b)

Figure 2: (a) The network of Fig. 1 with zero acknowledgement costs and residual node energies replaced by bandwidth constraints on the edges. There are several max-bandwidth $s - t'$ paths of cost 3. One such path is s, u, t, t', with bottleneck (u, t). (b) The max-bandwidth spanning tree computed by the modified Dijkstra's algorithm. Note that every path in the tree from s to a node w is a max-bandwidth $s - w$ path.

1. Transform $G = (V, E)$ into $G' = (V', E')$ by adding vertex t' and edge (t, t'). Set $b(u, v) = E_u - s_{u,v}$ for edges $(u, v) \in E$, and $b(t, t') = E_t$.
2. Use a max-bandwidth algorithm to find P', a max-bandwidth $s - t'$ path in G'.
3. Return P, the $s - t$ path in G corresponding to P'.

Figure 3: Algorithm for solving MREPP in networks without acknowledgement costs.

3.2 Solving MREPP

In this section we describe how to find optimal MRE paths when there are no acknowledgement costs for messages, i.e., $r_e = 0$ for all edges e. To solve this problem we run the max-bandwidth algorithm from the previous section on the graph obtained from the given network by defining the bandwidth of an edge as the residual energy that would remain in the node if we used the edge to route the message.

Assume we want to find an optimal $s - t$ path, where s and t are nodes of the given network $G = (V, E)$. Without acknowledgement costs, the residual energy of a node v_i on an $s - t$ path $P = (s = v_1, v_2, \ldots, v_{k-1}, v_k = t)$ is $R_{v_i} = E_{v_i} - s_{v_i, v_{i+1}}$, for $i = 1, \ldots, k - 1$, and it is $R_t = E_t$ for the node t. We now transform G into a new graph G' by adding a new node t' and a new edge (t, t') with bandwidth equal to E_t. For all other edges (u, v) of G' we define the bandwidth to be $b(u, v) = E_u - s_{u,v}$. Fig. 2(a) shows the transformed graph of Fig. 1 (when all acknowledgement costs are set to zero).

Obviously, there is a one-to-one correspondence between $s-t$ paths in G and $s-t'$ paths in G'. Moreover, if P is an $s-t$ path in G and P' is the corresponding $s-t'$ path in G' then $D(P) = B(P')$. For example, in Fig. 1 the path $P = (s, u, t)$ has $D(P) = 3$ (ignoring the acknowledgement costs), and the corresponding path $P' = (s, u, t, t')$ has $B(P') = 3$.

To find the max-bandwidth path in G', we can use the max-bandwidth algorithm from the previous section. The tree path from s to t' corresponds to the optimal MRE $s - t$ simple path in G. In Fig. 3 we summarize the algorithm to compute an optimal MRE path for networks without acknowledgement costs.

For example, running the modified Dijkstra algorithm on the transformed graph in Fig. 2(a),

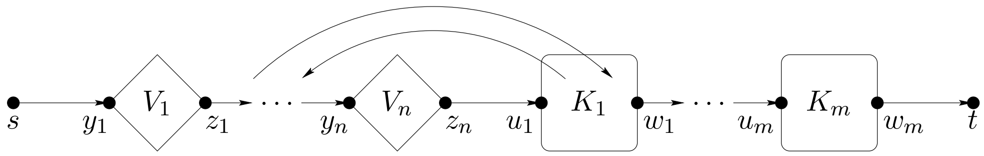


Figure 4: The NP-completeness reduction from 3-SAT. We try to send a message from s to t. This is possible if and only if the given 3-SAT formula is satisfiable. The connections between the gadgets V_i and K_j are only indicated.

gives the spanning tree in Fig. 2(b). In this tree, the max-bandwidth $s - t'$ path is $P' = (s, u, t, t')$ with bandwidth $B(P') = 3$. Thus, an optimal MRE $s - t$ path in the graph in Fig. 1 is $P = (s, u, t)$ with $D(P) = 3$ (ignoring acknowledgement costs).

4 The NP-Completeness Proof

In this section we show that MREPP is NP-complete. We actually show that even deciding whether there exists a legal routing path between two specified nodes is NP-complete. Later we will slightly modify the construction in the proof of the theorem to show that approximating MREPP is also very difficult.

We will use reduction from 3-SAT, so we first quickly review the 3-SAT problem [8]. An instance of 3-SAT is a set $U = \{x_1, x_2, \ldots, x_n\}$ of variables and a *formula* which is given as a collection $C_1, C_2, \ldots C_m \subseteq 2^U$ of *clauses*. A *literal* is either a variable x_i or its negation $\bar{x}_i$; in the former case the literal is *positive*, in the latter case it is *negative*. Each clause $C_j = \{\ell_{j,1}, \ell_{j,2}, \ell_{j,3}\}$ consists of three literals. A *truth assignment* is a function $\phi : U \to \{T, F\}$. A positive literal x_i is *satisfied* by ϕ if $\phi(x_i) = T$, a negative literal $\bar{x}_i$ is *satisfied* if $\phi(x_i) = F$. A clause is satisfied if at least one of its literals is satisfied. A formula is *satisfiable* if there is a truth assignment that satisfies all of its clauses. Deciding whether such an assignment exists is NP-complete.

Theorem 1 MREPP *is NP-complete.*

Proof. Since we can easily check whether a given routing leaves at least a specified minimum residual energy in the nodes along the routing path, MREPP is in NP.

We will prove that MREPP is NP-hard by showing a polynomial-time reduction from 3-SAT to MREPP. That is, given an instance of 3-SAT we will in polynomial time construct an instance of MREPP that has a legal path from s to t if and only if the 3-SAT instance is satisfiable.

We now describe how to transform an instance of 3-SAT to an instance of MREPP. Let the formula F be given as a set of m clauses $C_j = \{\ell_{j,1}, \ell_{j,2}, \ell_{j,3}\}$, for $j = 1, \ldots, m$, over n variables $x_1, \ldots, x_n$. We construct a directed graph as in Fig. 4. It contains two kinds of gadgets. For each variable x_i, $i = 1, \ldots, n$, we have the *variable gadget* V_i (see Fig. 5). In this gadget, the message can be routed along one of two parallel paths. If the message is routed along the upper path in the figure then we interpret this as setting x_i to T; therefore, we call this path the *true-path*. Similarly, the lower path corresponds to the setting $x_i = F$, so this path is called the *false-path*.

After the message has passed all the variable gadgets (and all variables have been assigned a truth value) it must be routed through the *clause gadgets* K_j, for $j = 1, \ldots, m$. To satisfy a clause, we must route the message through one of three parallel paths, each corresponding to one of the literals of the clause (see Fig. 6). Each of these paths moves back to a node in the variable gadget of the respective literal. If the clause contains the positive literal x_i then the clause gadget connects to the node $v_{i,j}$ on the false-path and if the clause contains the negative literal $\bar{x}_i$ then the clause

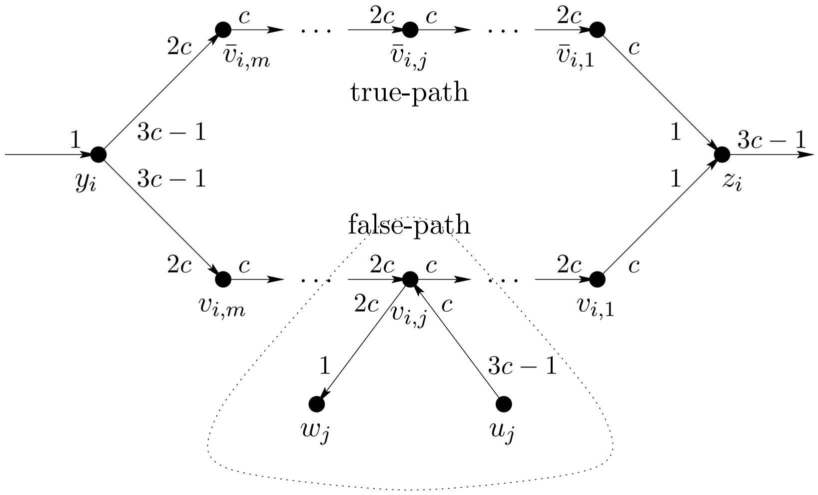


Figure 5: The variable gadget V_i. Routing the message along the upper (lower) path corresponds to setting x_i to T (F). The dotted line shows the interaction with clause C_j if that clause contains the literal x_i.

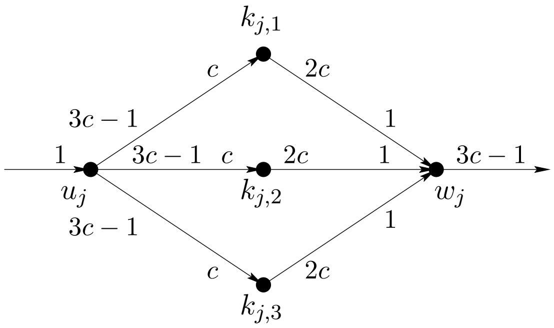


Figure 6: The clause gadget K_j. If $\ell_{j,p}$, for $p = 1, 2, 3$, is the variable x_i then $k_{j,p}$ is the node $v_{i,j}$ in V_i; otherwise, it is the node $\bar{v}_{i,j}$.

gadget connects to the node $\bar{v}_{i,j}$ on the true-path. Choosing the node energies and the send and acknowledgement costs appropriately (we will do this later), each node in a variable gadget can route the message at most once. Of course, if the formula is satisfiable then we can always route the message along the path corresponding to the literal in the clause that is true in the satisfying assignment. Thus, the existence of a satisfying assignment for the formula induces an $s - t$ routing path through the network. If all the nodes have sufficient energy for routing the message then this path is also legal. Fig. 7 shows the complete network for the formula $(x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3)$, together with the $s - t$ path corresponding to the satisfying truth assignment $x_1 = T$, $x_2 = F$, and $x_3 = T$.

It remains to show that every $s - t$ path induces a satisfying assignment for the formula. This can be guaranteed by a clever choice of the node energies and the send and acknowledgement costs for the edges. Let $c \geq 1$ be fixed. For the correctness of our reduction, the actual value of c is not important. But to get a polynomial-time reduction we must restrict c to a value that can be computed in polynomial time.

All nodes have energy $4c - 1$. The edge (s, y_1) has send cost $s_{s,y_1} = 3c$ and acknowledgement cost $r_{s,y_1} = 1$. The edge (w_m, t) also has send cost $s_{w_m,t} = 3c$ and acknowledgement cost $r_{w_m,t} = 1$.

In the variable gadgets V_i, for $i = 1, \ldots, n$, the incoming edge at y_i has acknowledgement cost

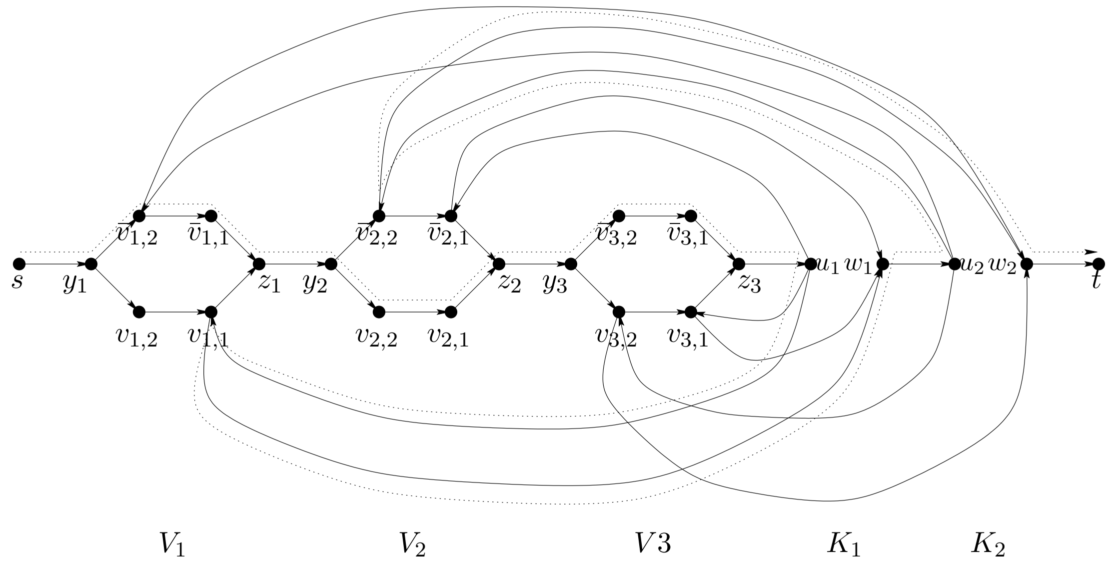


Figure 7: The network we construct for the formula $(x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3)$. The dotted line is the legal $s-t$ path corresponding to the satisfying truth assignment $x_1 = T$, $x_2 = F$, and $x_3 = T$.

1, and the outgoing edges of y_i have send cost $s_{y_i,\bar{v}_{i,m}} = s_{y_i,v_{i,m}} = 3c-1$ and acknowledgement cost $r_{y_i,\bar{v}_{i,m}} = r_{y_i,v_{i,m}} = 2c$. The incoming edges of z_i have send cost $s_{\bar{v}_{i,1},z_i} = s_{v_{i,1},z_i} = c$ and acknowledgement cost $r_{\bar{v}_{i,1},z_i} = r_{v_{i,1},z_i} = 1$, and the outgoing edge at z_i has send cost $3c-1$. All other edges on the true-path and the false-path have send cost $s_{\bar{v}_{i,j},\bar{v}_{i,j-1}} = s_{v_{i,j},v_{i,j-1}} = c$ and acknowledgement cost $r_{\bar{v}_{i,j},\bar{v}_{i,j-1}} = r_{v_{i,j},v_{i,j-1}} = 2c$, for $j = m,\ldots,2$. Thus, routing through V_i leaves residual energy $c-1$ in all nodes on the routing path (which is not enough to route the message a second time).

In the clause gadgets K_j, for $j = 1,\ldots,m$, the incoming edges of u_j have acknowledgement cost 1, and the outgoing edges have send cost $s_{u_j,k_{j,1}} = s_{u_j,k_{j,2}} = s_{u_j,k_{j,3}} = 3c-1$. The incoming edges at w_j have acknowledgement cost $r_{k_{j,1},w_j} = r_{k_{j,2},w_j} = r_{k_{j,3},w_j} = 1$, and the outgoing edges have send cost $3c-1$. Thus, routing through K_j leaves residual energy $c-1$ in all nodes on the routing path.

It is now easy to see that there is no legal $s-t$ path that does not correspond to a satisfying assignment. Consider an arbitrary legal $s-t$ path P. P must enter the gadget V_i at y_i, for $i = 1,\ldots,n$, follow either the true-path or the false-path (because the nodes do not have enough energy to send the message along one of the edges going to a node in a clause gadget), and leave V_i at node z_i. After the last gadget V_n, all nodes z_i have residual energy $c-1$, so P cannot pass through these nodes again later.

Now P enters the first clause gadget K_1. It can only follow an edge to a node v in some V_i that was not used before because the nodes in V_i do not have enough energy to receive and send two messages (or the same message twice). From v, which is either $v_{i,1}$ or $\bar{v}_{i,1}$, the message cannot be routed to z_i again (because the message passed already once through z_i). So it must go back to K_1 from v, i.e., it goes to w_1. But then P has satisfied clause C_1 because v corresponds to a true literal. Note that this argument is the reason why we ordered the nodes on the true-path (and false-path) in reverse order.

From w_1, P enters the second clause gadget K_2. Again, it must satisfy the clause by choosing an edge to either $v_{i,2}$ or $\bar{v}_{i,2}$ in some V_i that coresponds to a true literal. And again, it must immediately return to K_2 because otherwise it would have to leave V_i either via z_i or via w_1 (if x_1

is also a literal in C_1), and these both cases are not possible. Continuing this argument, we can prove by induction on $j = 1, \dots, m$ that P must enter K_j at u_j and leave it at w_j, so P indeed induces a satisfying assignment for all clauses. Also, all nodes on P will have energy $c-1$ after the routing, so we have $D(P) = c - 1$. Thus, P is legal if c is at least 1.

Finally, we must show that our reduction is polynomial-time. Our network has $O(nm)$ nodes and $O(nm)$ edges that can all be computed in polynomial time from the given description of the formula. Also, all edge costs and node energies can be computed in polynomial time if $c = O(2^{poly(nm)})$. Choosing $c = 1$ actually suffices for the correctness of the construction. We need other values of c later in the proof of non-approximability. □

In the proof of the theorem above, some of the edges have a higher acknowledgement than send cost. If this was not allowed in a more restrictive (but more realistic) routing scenario we could modify the construction by adding a new node with energy $4c-1$ in the middle of each edge, where the incoming edge has acknowledgement cost 1 and the outgoing edge has send cost $3c - 1$.

With another slight modification of the construction in the proof above we can also prove that approximating MREPP is very difficult.

Theorem 2 MREPP *cannot be approximated within a factor of* $O(2^{poly(n)})$ *in polynomial time unless* $P = NP$*, where* n *is the number of nodes in the network.*

Proof. We prove the theorem by showing that a polynomial-time approximation algorithm for MREPP with an approximation ratio of $\rho = O(2^{poly(n)})$ would enable us to solve 3-SAT in polynomial time. We use essentially the same construction as in the proof of Theorem 1. We just add one more edge from s to t, with send cost and acknowledgement cost $4c - 2$.

In this network, there always exists a legal $s - t$ path with minimum residual energy 1, namely the edge (s, t). Another path of minimum residual energy c exists if and only if the formula F represented by the network is satisfiable. Since we could choose $c = 2\rho = O(2^{poly(n)})$, any approximation algorithm for MREPP with approximation ρ could decide the satifiability of F. □

5 Conclusions

We have shown that MREPP is a very difficult problem in networks with acknowledgement costs. But this is only the problem of routing a single message. What we actually want is a good routing scheme for many messages, either in an online setting (and then we would like to use either competitive analysis or some probabilistic analyis) or in a situation where we have a batch of messages for which we want to find a good routing schedule. Since these problems seem to be rather hard, we might concentrate on finding heuristics that work well in practice.

References

[1] D. Bertsekas and R. Gallager. *Data Networks*. Prentice-Hall, 2nd edition, 1987.

[2] J.-H. Chang and L. Tassiulas. Routing for maximum system lifetime in wireless ad-hoc networks. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Sep. 1999.

[3] J.-H. Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *Proceedings of the 19th Annual IEEE Conference on Computer Communications (INFOCOM 2000)*, vol. 1, pp. 22–31, Tel Aviv, Israel, March 2000.

[4] J.-H. Chang and L. Tassiulas. Fast approximate algorithms for maximum lifetime routing in wireless ad-hoc networks. In Networking, vol. 1815 of Springer LNCS, pp. 702–713, 2000.

[5] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. McGraw-Hill and MIT Press, 1990.

[6] J. Edmonds and R. M. Karp, Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM* **19**(2): 248–264, 1972.

[7] M. Fredman and R. Tarjan. Fibonacci heaps and their uses in improved network optimization problems. *Journal of the ACM*, **34**: 596–615, 1987.

[8] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[9] J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian. Power aware routing in wireless packet networks. In *Proceedings of the Sixth International Workshop on Mobile Multimedia Communications (MoMuC'99)*, pp. 380-383, San Diego, California, Nov. 1999.

[10] J. C. Haartsen and S. Mattisson. Bluetooth — A new low-power radio interface providing short-range connectivity. *IEEE Proceedings*, **88**(10): 1651–1661, 2000.

[11] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. Chapter 5 in T. Imielinski and H. Korth, eds., *Mobile Computing*, pp. 153–181, Kluwer Academic Publishers, 1996.

[12] K. Kalyan Kumar and A. Chockalingam. *Energy efficient routing in wireless ad-hoc networks*. Url `http://citeseer.nj.nec.com/523912.html`.

[13] N. Malpani and J. Chen. A note on practical constructions of maximum bandwidth paths. Url `http://www.cs.tamu.edu/course-info/cpsc629/chen/notes/`, 2002.

[14] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and Applications*, **1**(2): 183–197, 1996.

[15] S. Singh, M. Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proceedings of the Fourth Annual International Conference on Mobile Computing and Networking (MOBICOM'98)*, pp. 181–190, Dallas, Texas, Oct. 1998.

[16] I. Stojmenovic and X. Lin. Power aware localized routing in wireless networks. In *Proceedings of the 14th IEEE International Parallel and Distributed Processing Symposium (IPDPS'00)*, pp. 371–376, Cancun, Mexico, May 2000.

[17] K. Woo, C. Yu, D. Lee, H. Y. Youn, and B. Lee. *Non-blocking, localized routing algorithm for balanced energy consumption in mobile ad hoc networks*. Url `http://citeseer.nj.nec.com/510965.html`.