Routing in Periodic Dynamic Networks using a Multi-Objective Evolutionary Algorithm

Uri Yael Rozenworcel[1]
[1] Universidad Católica *"Nuestra Señora de la Asunción"*
Lambaré – Central - Paraguay
+595 21 906066
tejuguasu@gmail.com

Benjamín Barán[1,2]
[2] Universidad Nacional de Asunción
Asunción - Paraguay
+595 21 585588
bbaran@pol.una.py

ABSTRACT

Dynamic networks are becoming very useful to model periodic behavior seen in sensor networks, low orbit satellites, ad hoc networks and other communication systems. To study those networks, the use of Evolving Graphs (EG) has several advantages. Mono-objective EG based algorithms as the *EG Foremost* and the *EG Shortest* were specially design to solve routing problems in dynamic networks. These algorithms clearly outperform state of the art routing approaches as *DSDV*, *DSR* and *AODV*; however, they only optimize one cost function as a time. On the contrary, this work models the problem as a purely Multi-Objective Optimization Problem (MOP) and proposes a Multi-Objective Evolutionary Algorithm (MOEA) to solve it. In fact, the implementation of a Strength Pareto Evolutionary Algorithm (SPEA) proved to be able to calculate a set of Pareto routes which contains solutions that are as good as the optimal ones calculated by the *EG Foremost* and the *EG Shortest* algorithms when considering a single objective, but may be even better when all cost functions are simultaneously considered. Experimental results confirm the advantage of using a MOEA for this problem.

Categories and Subject Descriptors

C.2.1 [**Computer – Communication Networks**]: Network Architecture and Design – *Network topology, Store and forward networks*.

C.2.2 [**Computer – Communication Networks**]: Network Protocols – *Routing protocols*.

General Terms

Algorithms, Measurement, Performance.

Keywords

Multi-Objective Problem (MOP), Multi-Objective Evolutionary Algorithm (MOEA) Strength Pareto Evolutionary Algorithm (SPEA), Periodic Dynamic Networks, Routing

1. INTRODUCTION

A dynamic network is composed by nodes which connect and disconnect from the network. The periods of disconnection could be in a planned, known or unknown way. The network could be centralized or ad-hoc [1, 2]. A path should be found to transmit a packet from a source node (remittent) to a destination (recipient). Internal nodes act as forwarders. If a node in the path is absent, the previous node should buffer the packet until the link to the node exists. A network is periodic if exists a period T such that at any instants t and $t + T$, the network has the same nodes and links attached, i.e., the same instant topology.

An *Evolving Graph* (EG) is a formalism that represents the connectivity state of a network at any instant [2]. It is composed by a set of sub graphs, one per *era*. An *era* is a period of time where the set of nodes and links that compose the network, stays unchanged. In this context, a *journey* is composed by ordered pairs of *hops* and *eras* in which the links should be used to transmit that packet.

Periodic dynamic networks are becoming very useful to model several communication networks. Nowadays, they are used to study Low Orbit Satellites (LEO), sensor networks [7], mobile ad-hoc networks (MANET) [4, 6] and mesh networks [3].

In what follows, Section 2 presents the problem in a multi-objective context, while the proposed approach to solve it is summarized in Section 3. The simulation environment is described in Section 4. Results for three networks are presented in Section 5, while the conclusions and future work are left for Section 6.

2. MULTI-OBJECTIVE OPTIMIZATION PROBLEM

The advantages of using Evolving Graphs (EG) to represent periodic dynamic networks were presented in [7]. Some known algorithms initially used for dynamic network routing are *Destination-Sequenced Distance-Vector* (DSDV) based on the Distributed Bellman-Ford algorithm [8], *Dynamic Source Routing* (DSR) [5] and *Ad-Hoc Distance-Vector* (AODV) [9].

In [7], Monteiro demonstrated that two mono-objective algorithms: (1) *EG Shortest* and (2) *EG Foremost,* are able to find an optimal solution in a mono-objective context, (i.e. equally or better than those found by the traditional algorithms above mentioned). The *EG Shortest* algorithm minimizes the hop count while the *EG Foremost* algorithm finds the earliest arriving era.

Until the present work, all published approaches treat the problem in a mono-objectives context, minimizing one of the following three journey measures: (1) the hop count, (2) the earliest era a packet arrives to the recipient or (3) the fastest a packet traverse the network, minimizing the number of eras it needs to travel [7]. Therefore, this work proposes to simultaneously consider all three objectives (cost functions) in a purely multi-objective context. Stating the routing problem as a Multiobjective Optimization Problem (MOP) allows finding a set of Pareto solutions that simultaneously optimizes the three objective functions which may be in conflict [10]; hence, compromised solutions are also of interest. Thus, a set of Pareto solutions may contain solutions that are as good as the optimal ones calculated by the *EG Foremost* and the *EG Shortest* algorithms when considering a single objective, but they may be even better when all cost functions are simultaneously considered, what is not done in a mono-objective context.

3. PROPOSAL OF A MULTI-OBJECTIVE EVOLUTIONARY ALGORITHM (MOEA)

Based on the Darwin and Mendel evolution's model, an Evolutionary Algorithm (EA) performs selective mating between individuals in order to evolve a set of solutions (known as population), hopefully generating better individuals at each generation. The individuals may be compared to each other applying Pareto Dominance to make sure that good solutions are calculated [10]. There are several good Multi-Objective Evolutionary Algorithms (MOEAs) that may be used for this problem [10], but only the Strength Pareto Evolutionary Algorithm (SPEA) will be reported in this article.

3.1 Strength Pareto Evolutionary Algorithm (SPEA)

The Strength Pareto Evolutionary Algorithm (SPEA) presented in [11] implements *elitism,* maintaining an external population of non dominated individuals (optimal solutions in a Pareto sense). The *strength* and *fitness* are calculated for each individual using Pareto dominance relations, as shown in [10]. *Selection* is implemented using a roulette involving both the actual evolving generation and the external population where best solutions are kept [10]. The probability of choosing an individual using the roulette is proportional to its *fitness* and therefore, to the number of individuals it dominates.

Let N be the set of nodes and E the set of eras in an evolving graph EG. Let n_s, $n_r \in N$, be the source and recipient nodes respectively. Let $e \in E$ be a particular era. $<n_s, n_r, e>$ represents a packet which is in n_s, and needs to be delivered to n_r being e the actual era in the *EG*.

Each individual contains information concerning how to route every possible packet $<n_s, n_r, e>$. This information is represented as a duple $<n', e'>$, $n' \in N$, $e' \in E$, where n' is the next node in the journey and e' the era when the packet should be sent.

In order to evaluate the performance of an individual, a packet is simulated to be sent from each possible source, to each possible recipient at each possible era. The objective functions calculated with the simulation are: (a) the sum of hops needed to deliver all the packets; (b) the sum of eras that the packets have lived since creation until recipients receive each transmitted packet (i.e. packets lived eras); and (c) the sum of eras spent by the packets traversing the network since being first transmitted by the source (i.e. traveling eras). The shortest journey has a minimum sum of hops, minimizing (a); the foremost journey has a minimum sum of lived eras, minimizing (b), while the fastest journey has a minimum sum of traveled eras, what minimizes (c).

An individual is said to be *valid* if regarding the information it contains, all possible packets are assured to reach the recipients. At every generation, each individual goes through a crossover and mutation processes used to obtain new evolved solutions. As a result of those processes, the routing information in those new individuals could contain unnecessary loops in the journey, or could fragment the network making impossible to reach a recipient. Therefore, for a MOEA to be really useful, new individuals should be fixed to be valid solutions. Consequently, the present work proposes a novel method called *reformatory* to repair a *new individual* to obtain a *valid individual* that really is a solution of the routing problem in the studied EG.

3.2 Reformatory: Convert a New Individual into a Valid Solution

Given the routing tables for each era that composes an individual, all possible combinations of $<n_s, n_r, e>$ packets are sent, one at a time. Two sorted lists are maintained: one records the visited nodes denoted as **`taboo`**, so a loop in the journey would be discovered; the other list **`taboo_n`** tracks, for each node, which pairs of adjacent nodes and eras $<n', e'>$ were already tried as next hop in the journey to the recipient. A partition of the network is detected if every possible neighbor (adjacent node) at every possible era, is present in **`taboo_n`** (see Algorithm 1).

When either a loop or a partition is detected, the algorithm performs a rollback and it modifies the routing table to fix the problem. Randomly, a new hopping alternative is taken from the *EG Shortest* routing table, from the *EG Foremost* one [1] or from a neighbor and an era $<n', e'>$ not present in **`taboo_n`**.

Initially, all routing information is market as *`test`* to make sure that it will be checked by Algorithm 1 that has the job of making sure that every packet can be correctly delivered to destination. If this is not the case, it systematically changes routing information and marks it as *`suggested`*, until every packet gets to its recipient, marking at this time each checked entry in the routing table as *`correct`*. In summary, Algorithm 1 receives a possible routing solution (individual) and returns a verified correct solution of the problem.

4. SIMULATIONS

To make sure that a MOEA is a good alternative to solve periodic dynamic networks, extensive simulations were performed to compare it to other known algorithms as *DSDV*, *DSR*, *AODV*, as well as optimal mono-objective algorithms as *EG Shortest* and *EG Foremost*.

To have an idea of the performance of the above algorithms, three different *scenarios* are reported in this paper. A scenario is composed by a network topology giving nodes, links and eras information as well as a traffic demand with a detailed information of each packet to be transmitted, as explained below.

```
//Let taboo be an ordered list of nodes n, already visited in a journey
//Let taboo_n[n] be the set of pairs <n', e> where node n' is adjacent to node n,
//and e represents an era already tried.
//Let n_s, n_r be the sender and recipient nodes, e_sr be the era when the packet is available to be sent
//Let <n_s, n_r, e_sr> be a packet to be sent from n_s to n_r while the simulation is in era e_sr
//Let ind[<n_s,n_r,e_sr>] be an index that point to the information <n,e> for the next hop
1. For each pair of nodes and for each era, mark all routing information as test
2. While there exists information not marked as correct do
2.1. For each pair of nodes and each era <n_s, n_r, e> do
2.1.1. While n ≠ n_r
2.1.1.1. If ind[<n, n_r, e>] is marked as test
2.1.1.1.1. Mark the information of ind[<n, n_r, e>] as suggested
2.1.1.1.2. <n', e'> ← link indicated by ind[<n, n_r, e>]
2.1.1.1.3. If taboo includes n', or taboo_n[n] includes <n',e'>, then
           perform rollback and try another path
2.1.1.2. If ind[<n, n_r, e>] is marked as correct
2.1.1.2.1. If taboo includes n', or taboo_n[n] includes <n',e'>, then
           perform rollback and try another path
2.1.1.3. If ind[<n, n_r, e>] is marked as suggested
2.1.1.3.1. Calculate the set of adjacent nodes to n that are not in taboo, and the available eras
           that are not in taboo_n[n]
2.1.1.3.2. If there are no adjacent nodes and available eras, then
           perform rollback and try another path
2.1.1.3.3. Choose randomly one of the three possibilities: get the information for ind[<n, n_r, e>]
           from a known routing coming from EG Shortest, EG Foremost, or randomly get a link
2.1.1.4. Add n to taboo and add ind[<n, n_r, e>] to taboo_n[n]
2.1.1.5. <n,e> ← ind[<n, n_r, e>]
2.2. For each <n, n_r, e> of the journey, mark the information as correct
```

Algorithm 1. Pseudocode of the Reformatory procedure

4.1 Network Topology

A network topology specifies every node $n \in N$, and every link (n_i, n_j) at each era *e*. Three network sizes are reported in this work (see Figures 1, 2 and 3). For all the reported simulations, only 15 eras were considered. The smallest network with five nodes and seven links is presented in Figure 1 where the *eras* at which each link exists is presented between brackets on top of each link. For example, link (A, C) of Figure 1 only can be used at eras 1, 7 and 13, i.e., (A, B) only works in three out of 15 eras.

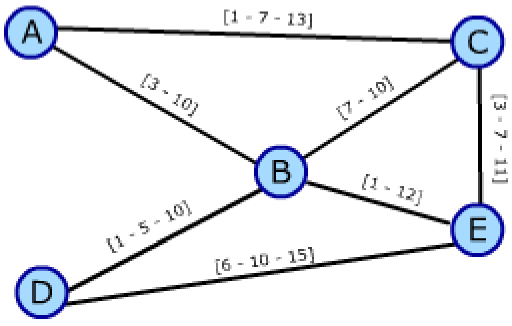

Figure 1. An EG representing a small sized network

Based on the small topology of Figure 1, a more complex topology is presented in Figure 2. This medium size network is formed by thirteen nodes and twenty links.

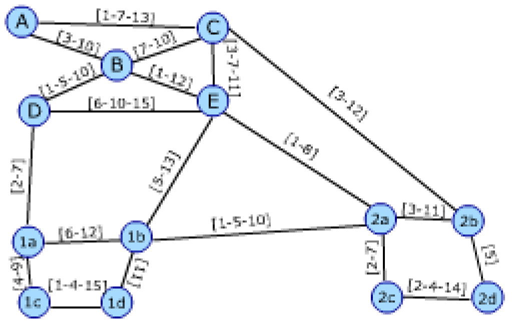

Figure 2. An EG representing a medium sized network

A third more complex network with twenty-four nodes and thirty seven links (see Figure 3) is also simulated in the present work.

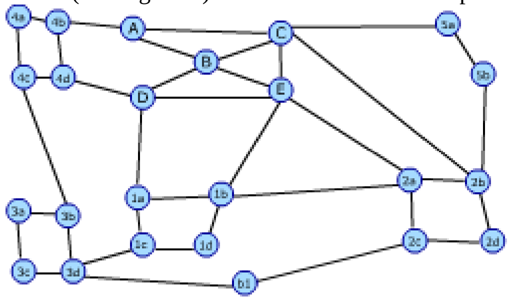

Figure 3. An EG representing a large sized network

Given the amount of information needed to detail all the information on which eras each link is available, it is presented in Table 1 instead of including the information in Figure 3.

Table 1. Active eras for each link in Figure 3.

Node	Node	Eras
A	B	3-5
A	C	1-2-4
B	C	1-3-5
B	D	1-2-3
B	E	3-4-5
C	E	1-4-5
D	E	1-2-4
D	1a	1-3-5
E	1b	1-2-4
1a	1b	3-5
1a	1c	2-3
1b	1d	5
1c	1d	3-4-5
2a	E	1-3-5
2b	C	1-2-4
2a	2c	1-2-3
2a	1b	1-2-3-4-5
2a	2b	3-5
2b	2d	5

Node	Node	Eras
2c	2d	3-4-5
3a	3c	2-3
3a	3b	3-5
3b	3d	5
3c	3d	3-4-5
3d	1c	3-4-5
4a	4b	3-5
4a	4c	2-3
4b	4d	5
4b	A	3-4-5
4c	4d	3-4
4d	D	4-5
4c	3b	2-4
b1	3d	1-2-3-4-5
b1	2c	1-2-3-4-5
5a	5b	1-2-3-4-5
5a	C	3-5
5b	2b	2-4

4.2 Traffic Load

Traffic is the load specification of a network, i.e. the exact description of each packet to be transmitted. A packet is represented as a tuple <n_s, n_r, e>, where n_s and n_r are the sender and recipient nodes. The packet *is born* at era *e*, and it is scheduled to be sent according to the node's routing table.

For space reasons, this work only presents simulations using heavy loaded networks. The number of packets to be transmitted in the simulations is calculated as 90 % of the number of nodes multiplied by the number of links that constitute the network. As an example, Table 2 summarizes the 31 packets randomly chosen for the first simulation with the smallest network shown in Figure 2. Traffic load in a congested medium sized network considers 234 packets (for Figure 2), while 779 packets are considered for the largest network of Figure 3.

Table 2. Traffic Load for the Topology of Figure 1

Source	Recipient	Era
A	D	1
A	D	2
A	D	3
A	D	4
A	D	5
A	D	6
A	D	7
A	D	8
A	D	9
A	D	10
A	D	11
B	E	4
B	E	6
B	E	7
B	E	8
B	E	9

Source	Recipient	Era
B	E	10
B	E	11
B	E	12
B	E	13
B	E	14
E	A	1
E	A	3
E	A	5
E	A	7
E	A	8
E	A	11
E	A	12
E	A	13
E	A	14
E	A	15

4.3 SPEA Parameters

To run the Strength Pareto Evolutionary Algorithm with the special adaptation for this problem, as the reformatory presented in section 3.2, some parameters should be defined. Table 3 shows an example of the value for each parameter used to solve one instance of the smallest problem presented in Figure 1.

In Table 3, P_{max} represents the constant size of the evolutionary population at each generation, while P'_{max} represents the maximum allowed size for the external population where non-dominated optimal solutions are stored. Th number of iterations is given by I_{max}, naming each iteration as I_k.

To optimize in a multiobjective context the solutions found by the optimal mono-objective algorithms *EG Shortest* and *EG Foremost*, they may be inserted in the evolutionary population *P* after ***EG insertion*** iterations. That way, better solutions may be easily found when all three objective functions are simultaneously considered. The insertion of the *EG Foremost* and *EG Shortest* solutions [1] into the external population *P'* has the effect of speeding up the search of non dominated solutions and, when possible, dominate them.

When the Reformatory procedure presented in Section 3.2 finds a problem in an individual routing table, it tries to fix it changed the problematic value. To do so, it consults the routing table found by the *EG Shortest* with probability ***P_EG_f***, or it looks for an entry in the routing table calculated with *EG Foremost,* with probability ***P_EG_s.*** Alternatively, it randomly takes a valid neighbor in a valid era with probability ***P_rand.*** Therefore, the reformatory randomly decides a strategy to fix routing information, with probabilities shown in Table 3.

Table 3. Example parameters used in SPEA.

Size	P_{max}	I_{max}	P'_{max}	EG insertion	P_EG f	P_EG s	P_rand
Small	10	1000	10000	1	0,4	0,4	0,2

5. RESULTS

In all tests performed, the *SPEA* always found very good solutions at the prize of running time. The number of solutions depends mainly on the population size and the number of iterations as will be presented in Table 9.

The following sections present experimental results for the three topologies shown in Figures 1, 2 and 3, obtained using the following 6 algorithms:

1. *Destination-Sequenced Distance-Vector (DSDV).*
2. *Dynamic Source Routing* (DSR).
3. *Ad-Hoc Distance-Vector* (AODV).
4. *EG shortest (EG s).*
5. *EG foremost (EG f).*
6. *Strength Pareto Evolutionary Algorithm (SPEA).*

5.1 Simulations with a Small Network

Simulations performed with the network presented in Figure 1 have resulted in the solutions shown in Figure 4. To simplify the reader understanding, Figure 4 shows an average of only two objective functions: (a) hop count and (b) packet lived eras. The third objective function (c) traveling eras, is not shown for simplicity but it may be appreciated given the large number of compromised solutions found by the SPEA algorithm, using parameters presented in Table 3. A legend in the right side of Figure 4 indicates which algorithms were tested.

Table 4 presents a very short summary of the 3 objective functions calculated by the *EG shortest*, *EG foremost* and two out of 168 SPEA solutions. As can be read in the comment column of the table, a solution calculated with SPEA, denoted as SPEA 1 dominates the solution found using *EG foremost*, given that both have the same average eras to destination (of 2.61) but the SPEA 1 solution has smaller values in the other two objective functions. At the same time, another solution calculated using SPEA, denoted as SPEA 2, dominates the solution found using *EG shortest*, given that both need an average of 1.68 hops to arrive, but SPEA 2 needs in average a fewer number of eras. Remember that *EG foremost* and *EG shortest* calculate optimal solutions in a mono-objective context [1]; therefore, it is remarkable that SPEA can find better solutions than those EG algorithms.

Table 4. Average objective functions calculated with solutions obtained by algorithms SPEA, EG shortest and EG foremost, for a small network.

Algorithm	(a) Hop count	(b) Eras to destination	(c) Eras traveling	Comment
EG foremost	2.03	2.61	1.32	dominated by SPEA 1
EG shortest	1.68	4.07	1.16	dominated by SPEA 2
SPEA 1	1.90	2.61	0.94	dominates *EG f*
SPEA 2	1.68	3.55	0.90	dominates *EG s*

Another advantage of the SPEA algorithm proposed in this work is its ability to find a whole set of Pareto solutions that may be used to decide which solution best fit the routing necessity of the users. As an example, for this small network, 168 solutions were found, as shown in the Figure 4 as dashes.

For completeness, it should be mentioned that *EG Foremost* and *EG Shortest* always dominate *DSDV*, *DSR* and *AODV* solutions as reported in [1]. In turns, *SPEA* is able to calculate solutions that dominate all of them. This fact is emphasized in the following subsections where the named algorithms are compared in each measured domain. Three individuals among the 168 SPEA's solution set were selected to facilitate comparison. They are denoted as: SPEA 1, SPEA 2 and SPEA 3.

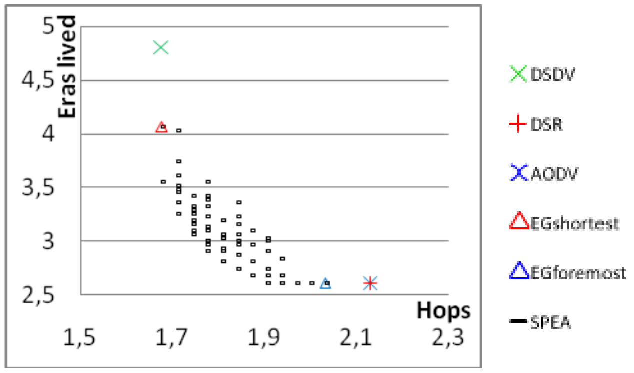

Figure 4. Example set of solutions for a small sized network.

5.1.1 *Journey Hops*

As expected, the *EG shortest* algorithm always found the shortest journey to deliver a packet. Figure 5 shows how many hops it takes for each of the thirty-one packets to be delivered to the recipient. None of the other algorithms named in the legend's graph performed better considering only this objective function (hop count). Because *EG Shortest* is an adaptation of Dijkstra's Shortest Path Algorithm [1], it is said to be optimal.

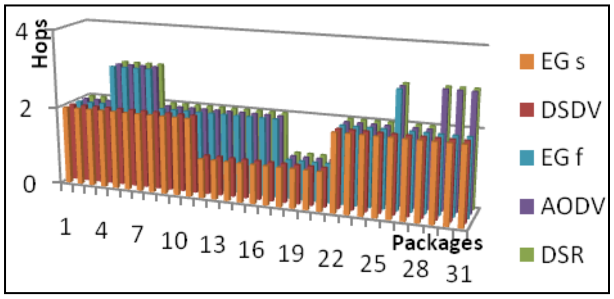

Figure 5. Hops taken for each packet to reach the recipient.

The *SPEA* found a solution, named *SPEA 2*, which performs as well as *EG Shortest* considering the hop count (see Figure 6). However, as shown in Table 4, it performs better than *EG Shortest* making the journeys faster in average and reducing the packets life, so SPEA 2 is a better solution than the one calculated by *EG Shortest*.

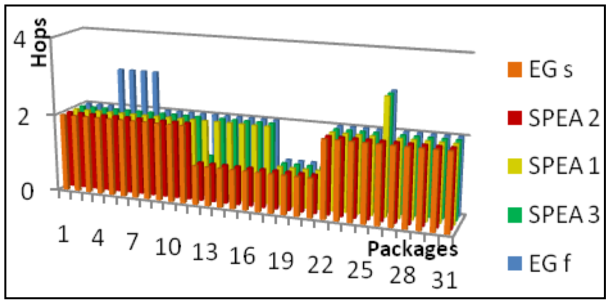

Figure 6. Hops count routing with SPEA solutions.

5.1.2 *Eras to Destination*

A packet is born when information is ready to be delivered at the sender node. The packet's life is considered to be the elapsed number of eras between the era in which it is born and the era in which it reaches the receiver node.

The algorithm *EG Foremost* is an adaptation of Dijkstra's Shortest Path Algorithm but it minimizes the packet's life rather than the journey's length [1]. It's said to be optimal when only this objective function is considered.

Figure 7 shows a comparison among the 5 mono-objective algorithms listed in the graph's legend. As expected, the *EG Foremost* performs better than *EG Shortest* and *DSDV*, but it is interesting to note that it performs equally well as *DSR* and *AODV* for the topology given in Figure 1.

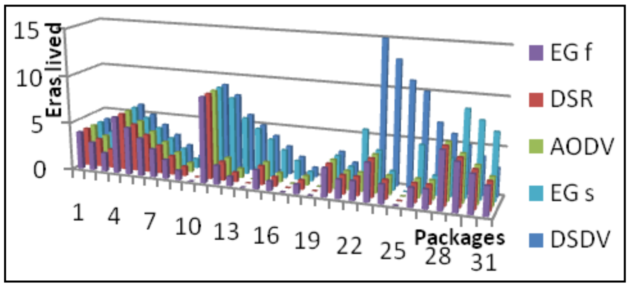

Figure 7. Life of each packet for mono-objective algorithms.

The *SPEA* found a solution, named *SPEA 1*, which performs as well as *EG Foremost* when only the average number of eras to destinations is considered (see Figure 8). However, as shown in Table 4, it performs better than *EG Foremost* when the other two objective functions are considered, making the journeys faster and reducing the hop count average in the journeys, so SPEA 1 is a better solution than the one proposed by *EG Foremost* algorithm.

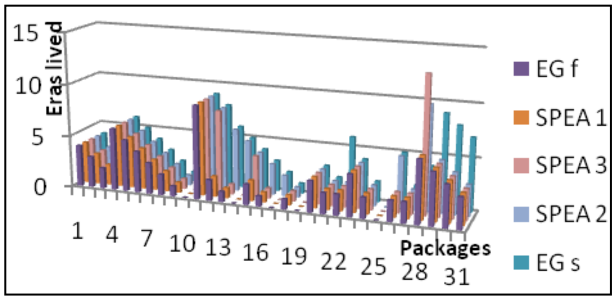

Figure 8. Life of each packet for different algorithms.

5.1.3 Traveled Eras

A journey is faster than other if it takes a smaller number of eras traversing the route until the recipient. The traveled eras count is calculated as the arriving era minus the era a packet leaves for the first time its source. If the packet goes from sender to recipient in the same era, it is said to be a journey of cero eras.

An algorithm, *EG Fastest*, could be an adaptation of Dijkstra's Shortest Path Algorithm, minimizing the number of eras that a packet takes to be routed from sender to recipient [1]. Until the present publication it was not implemented nor published. As in previews subsections, *EG Fastest* would find the optimal journey under a mono-objective framework but it might be dominated by a solution calculated by the SPEA algorithm.

The three *SPEA*'s solutions of Figure 9 perform better than the algorithms *EG Shortest*, *EG Foremost*, *DSDV*, *DSR* and *AODV*.

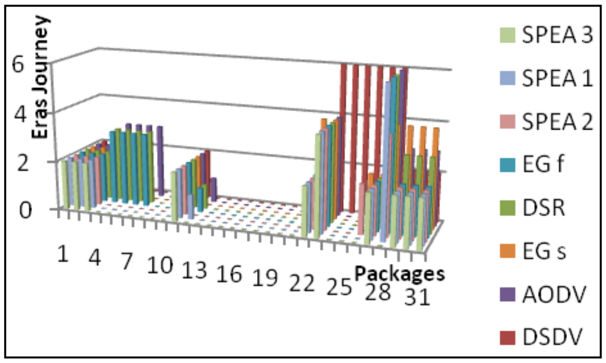

Figure 9. Number of traveling eras for each packet using different algorithms.

5.2 Simulations with a Medium Network

Simulations performed with the medium sized network of Figure 2, is summarized in the Figure 10, where *SPEA* used the parameters shown in Table 5.

Table 5. Example parameters used in SPEA.

Size	P_{max}	I_{max}	P'_{max}	EG insertion	P_EG f	P_EG s	P_rand
Medium	10	1000	10000	1	0,3	0,3	0,4

Table 6 presents the average values of the 3 objective functions calculated by the *EG shortest*, *EG foremost* and two out of 53 SPEA solutions, denoted as *SPEA' 1* and *SPEA' 2*. In this case, the solutions *SPEA'2* and *EG Foremost* have the same objective function values. Both solutions *SPEA' 1* and *EG Shortest* have the same hop count value (of 2.927), but the *SPEA' 1* has smaller values in the eras to destination and eras traveling objective functions.

Table 6. Average objective functions calculated with solutions obtained by algorithms SPEA, EG shortest and EG foremost, for a medium network.

Algorithm	(a) Hop count	(b) Eras to destination	(c) Eras traveling	Comment
EG foremost	3.526	6.482	4.897	
EG shortest	2.927	10.667	7.449	dominated by *SPEA' 1*
SPEA' 1	2.927	10.517	7.376	dominates *EG f*
SPEA' 2	3.526	6.482	4.897	

Figure 10 shows the solutions of the algorithms named in the legend's graph. The set of 53 non dominated solutions calculated by *SPEA* includes the solution found by *EG Foremost* and a solution that dominates the one found by *EG Shortest*, outperforming it once more (see Table 6).

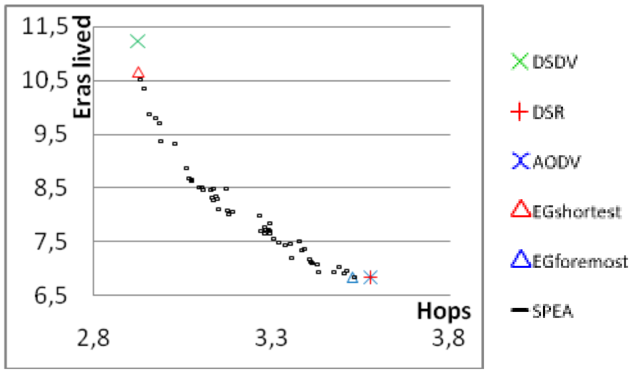

Figure 10. Example set of solutions for a medium sized network.

5.3 Simulations with a Large Network

A network composed by twenty-four nodes was presented in Figure 3. Using the parameters showed in Table 7 and traffic of 630 packets, a test was performed in the same way as the above described. Experimental results are presented in Figure 11.

Table 7. Example parameters used in SPEA.

Size	P_{max}	I_{max}	P'_{max}	EG insertion	P_EGf	P_EGs	P_rand
Large	10	100	10000	1	0,3	0,3	0,4

The algorithm *EG Shortest* performed as well as the *DSDV* in the hop count measure. The algorithm *EG Foremost* exceeded *DSR* and *AODV* performance in the packet life's measure.

In this case, the *SPEA* approach found 22 non dominated solutions, three of them denoted as SPEA'' 1, SPEA'' 2 and SPEA'' 3.

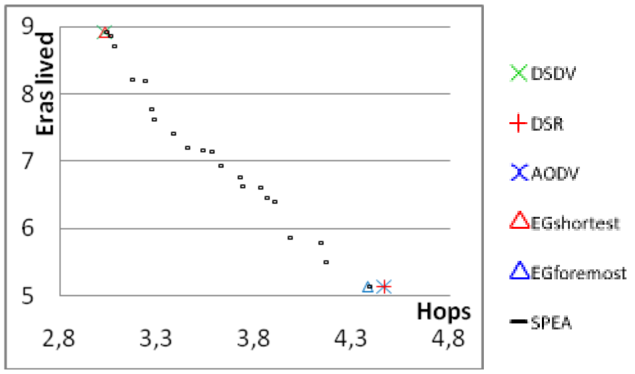

Figure 11. Example set of solutions for a large sized network.

The objective functions average values are briefly summarized in Table 8. The solution set contain those proposed by the *EG Shortest* and *EG Foremost* algorithms (SPEA'' 1 and SPEA'' 2), and also contains solutions (SPEA'' 3) that responds to a trade-off among the three different objective functions: (a) hop count, (b) packets life and (c) traveled eras.

Table 8. Average objective functions calculated with solutions obtained by algorithms SPEA, EG shortest and EG foremost, for a large network.

Algorithm	(a) Hop count	(b) Eras to destination	(c) Eras traveling
EG foremost	4.379	0.548	4.897
EG shortest	3.033	8.927	3.27
SPEA'' 1	3.033	8.927	3.27
SPEA'' 2	4.379	0.548	4.897
SPEA'' 3	3.443	1.825	7.203

5.4 Variation of SPEA Parameters

The observations asserted in the present section, are based on an exhaustive testing over the small network of Figure 1 and the same traffic definition of thirty one packets presented in Table 2. Experimental results are presented in Table 9 for different values of the used parameter.

Table 9 is similar to Tables 3, 5 and 6, but it also includes other columns defined below. **Solutions** gives the number of individuals that compose the Pareto front calculated by the *SPEA* after I_{max} iterations; **Non Dominated Solutions** is the number of solutions found by the *SPEA* that are not dominated by the any solution found by other algorithms as *DSDV*, *DSR*, *AODV*, *EG Shortest* and *EG Foremost*. **Dominates EGf** and **Dominates EGs** indicate whereas there exist at least one solution found by the SPEA which performs better than the solution calculated by *EG Foremost* or *EG Shortest*, respectively.

Each test instance will be referred by an index in the first column of Table 9 to facilitate references. A set of instances indexed by the first column will be presented between curly brackets. For example, instances 7, 8 and 9 will be denoted as {7, 8, 9}.

Finally, it is worth saying that the number of **solutions** found by the *SPEA* in its successive iterations, does not assure quality. In contrast, the valuable ones are the **non dominated solutions**, given that they are part of the Pareto front with the non dominated solutions calculated with all the other algorithms, if they exist.

5.4.1 Size of the population

When maintaining constant the other parameters, increasing the size of the population improves exploration and therefore, the number of *non dominated solutions*. As an example, consider the set of tests {7, 8, 9} of Table 9 when SPEA varies its population size and notably improves the quality of the found solutions. Although {9} has fewer solutions in its Pareto front than {8}, the quality of the solutions are clearly better, which could be seen from the enormous improvement in the number of non dominated solutions.

The magnitude of improvement decreases when *EG insertion* is not set to cero (e.g. like in {10, 11, 12}), or when the external population includes solutions that dominates the *EG Shortest* or *EG Foremost* ones.

Table 9. Tests over a small sized network varying parameters.

Nº	*P_{max}*	*I_{max}*	*EG insertion*	*P_EGf*	*P_EGs*	*P_rand*	*Number of Solutions*	*Non Dominated Solutions*	*Dominates EGf*	*Dominates EGs*
1	20	100	0	0,3	0,3	0,4	15	1	No	No
2	10	100	0	0,4	0,4	0,2	2	0	No	No
3	10	1000	0	0,4	0,4	0,2	51	49	No	No
4	10	10000	0	0,4	0,4	0,2	9999	2105	No	No
5	20	100	0	0,4	0,4	0,2	12	0	No	No
6	50	100	0	0,4	0,4	0,2	11	8	No	No
7	10	100	0	0,2	0,2	0,6	3	0	No	No
8	20	100	0	0,2	0,2	0,6	12	1	No	No
9	50	100	0	0,2	0,2	0,6	10	10	No	No
10	10	100	1	0,3	0,3	0,4	82	22	Yes	Yes
11	20	100	1	0,3	0,3	0,4	106	25	Yes	Yes
12	50	100	1	0,3	0,3	0,4	167	24	Yes	Yes
13	10	100	50	0,3	0,3	0,4	77	32	Yes	No
14	20	100	50	0,4	0,4	0,2	86	21	No	Yes
15	10	100	1	0,4	0,4	0,2	94	15	Yes	No
16	10	100	1	0,2	0,2	0,6	88	19	Yes	Yes
17	50	100	1	0,4	0,4	0,2	148	56	Yes	Yes
18	50	100	1	0,2	0,2	0,6	142	40	No	No

5.4.2 Number of iterations

Increasing the number of iterations has a direct influence over the number of non dominated solutions found, as can be seen in {2, 3, 4}. Therefore, increasing the number of iterations has a similar effect than incrementing the size of the population (see section 5.4.1) in the non dominated solutions column.

5.4.3 Probabilities at the reformatory

In general, simulations used the same probabilities for ***P_EGf*** and ***P_EGs***, thus, to avoid possible bias. Most of the times, the best experimental results were obtained with ***P_EGf*** = ***P_EGs*** = 0.3 and ***P_rand*** = 0,4, as can be seen when considering the instance sets {10, 15, 16} and {1, 5, 8} from Table 9.

Nevertheless, when the set {12, 17, 18} is considered it is possible to see that not always the best distribution is the one described above.

6. CONCLUSIONS AND FUTURE WORK

Given that dynamic networks are becoming very useful to model periodic behavior seen in sensor networks, low orbit satellites, ad hoc networks and other communication systems, it is very useful to study those networks using Evolving Graphs (EG), as demonstrated in [1]. Up to now, this problem was only studied considering a single objective function as (a) the sum of hops needed to deliver all the packets; (b) the sum of eras that the packets have lived since creation until recipients receive each transmitted packet; and (c) the sum of eras spent by the packets traversing the network since being first transmitted by the source.

This work models the above problem as a purely multiobjective problem (MOP), for the first time. To solve this multi-criteria problem, a Multi-Objective Evolutionary Algorithm (MOEA) was developed inspired in the Strength Pareto Evolutionary Algorithm (SPEA), which was adapted to the specific problem with specialized procedures as the *Reformatory* described in section 3.2. In this way, the proposed approach was able to find a whole set of Pareto solutions that not only contains individuals competitive with state of the art algorithms, but also is able to calculate individuals that dominate solutions considered optimal when only a single objective function is considered, given that the proposed approach is able to improve in a multiobjective context solutions that may be considered optimal in a single-objective approach. It is also important to remark the advantage of calculating a whole set of trade-off solutions in only one run of the algorithm, giving the decision maker the ability to choose the alternative that best fit his needs.

Future work could include deviations in the planned behavior of each node, using a predefined EG as a guideline or estimation of the network state at each era. Quality of service may also be considered as well as the alternatives to avoid congestion, managing flow control mechanisms.

7. REFERENCES

[1] Ash G.R. 1995. Dynamic Network Evolution, with Examples from AT&TS Evolving Dynamic Network. IEEE Communication Magazine, vol. 33 Nº7, July 1995 pp. 26-39. http://dl.comsoc.org/cocoon/comsoc/servlets/GetPublication?id=111660.

[2] Bui Xuan, B., Ferreira, A. and Jarry A. Computing Shortest, Fastest, and Foremost Journeys in Dynamic Networks. International Journal of Foundations of Computer Science, Volume 14, Number 3, June 2003, pp. 267-285.

[3] Campanha, R., Bletsas, M., Claudio, L. and Magalhães, S. Mesh Networks for Digital Inclusion - Testing OLPC's XO Mesh Implementation. In: 8o Forum Internacional de Software Livre, 2007, Porto Alegre. Anais da Trilha Internacional do Workshop do 8 Forum Internacional de Software Livre. 2007.

http://www.midiacom.uff.br/~schara/publications/wsl2007.pdf.

[4] Jarry A. and Lotker, Z. Connectivity in Evolving Graph with Geometric Properties. October 2004. DIALM-POMC '04: Proceedings of the 2004 joint workshop on Foundations of mobile computing. Philadelphia, PA, USA, pp. 24-30.

[5] Johnson D.B. and Maltz, D.A. Dynamic Source Routing in Ad Hoc Wireless Networks. Imielinski and Korth, editors, Mobile Computing, vol. 353, pp. 153–181. Kluwer Academic Publishers, 1996.

[6] Kastner W. and Leupold, M. How dynamic networks work: A short tutorial on spontaneous networks. In Proc. 8th IEEE Intl. Conference on Emerging Technologies and Factory Automation (ETFA '01), Vol. 1, pp. 295-303, 2001. http://ieeexplore.ieee.org/search/wrapper.jsp?arnumber=996382.

[7] Monteiro, J. June 2007. Uso de Grafos Evolutivos no Roteamento em Redes Dinâmicas: algoritmos, fluxos e limites. Masterly thesis. Instituto de Matemática e Estatística, Universidade de Sâo Paulo. http://doi.acm.org/10.1145/161468.161471.

[8] Perkins, C. and Bhagwat, P. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In ACM SIGCOMM'94: Proceedings of the Conference on Communications Architectures, Protocols and Applications. ACM Press, October 1994, pp. 234-244. ISBN:0-89791-682-4.

[9] Tomar, G.S. Modified Routing Algorithm for AODV in Constrained Conditions. Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on Volume , Issue , 13-15 May 2008 pp. 219–224. Digital Object Identifier 10.1109/AMS.2008.196.

[10] von Lücken, C. Algoritmos Evolutivos para Optimización Multiobjetivo: un estudio comparativo en un ambiente paralelo asíncrono. Masterly thesis. Universidad Nacional de Asunción. December 2003.

[11] Zitzler, E. and Thiele, L. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, Noviembre 1999.