Evolving greenfield passive optical networks

E Bonsma, N Karunatillake, R Shipman, M Shackleton and D Mortimore†

We investigate applying an evolutionary algorithm (EA) to the design of a passive optical network (PON). We use three techniques to improve the performance. Firstly, to reduce the risk of sub-optimal convergence, we use a novel genetic encoding. Secondly, we combine the EA with a heuristic to guide the optimisation. Thirdly, we investigate various ways of sub-dividing the problem. We briefly present experiments to demonstrate how the EA performs. The results show the strengths and weaknesses of the various techniques we employ.

1. Introduction

The design of a telecommunications network is often a difficult problem. The many interdependent choices that need to be made, the conflicting criteria that apply and the constraints that need to be obeyed result in a search space that is typically very large and complex. Evolutionary computation is an optimisation technique that is often very effective in this case. Evolutionary algorithms (EAs) are therefore frequently applied to the design of telecommunications networks [1, 2].

In this paper we look at an idealised version of a real world network design problem, the design of a passive optical network (PON). A PON is a point-to-multipoint access network that uses low-cost, passive splitters. The problem has a large and complex search space which makes the optimisation difficult. However, it abstracts from specific details because our focus here is on general techniques that can be used to improve the performance of EAs when applied to real-world problems. The three techniques we consider are the use of a neutral encoding, the combination of the EA with problem-specific heuristics, and splitting the optimisation into several stages.

The rest of this paper is structured as follows. Section 2 describes the network design problem in detail. Section 3 then discusses the implementation of the EA, with particular focus on the above three techniques. Section 4 briefly presents some experimental results that show how the EA performs. Finally, section 5 draws conclusions.

2. Problem description

The problem we consider is a generalised version of the greenfield design of a PON. Greenfield means that the network is developed from scratch.

† Evolved Networks

There are a number of customers, each at a separate location with a demand of one or more network connections, potentially of different types, e.g. four LAN and two DS1. Each customer needs to be connected to a host, which can be shared by multiple customers. The host connects customers to the optical network by way of service boxes that are allocated to the host. There are different configurations of service boxes, each able to handle a certain number of customer connections, see Table 1. Hosts are connected by way of optical fibre to a splitter, which can be shared by multiple hosts. Each splitter is then connected to an exchange. Figure 1 shows a very simple example network. As shown, a customer can be collocated with a potential host.

Table 1 Example service box types.

Service box	Cost
5 LAN	200
5 DS1	200
5 LAN + 5 DS1	350
10 LAN	325
10 DS1	325
10 LAN + 10 DS1	600

The problem specification includes the locations for customers and the potential locations for hosts, splitters and exchanges. There are costs associated with the use of each host, splitter and exchange, which may depend on the location. For each host, there is also a limit on the number of service boxes that can be allocated to it. There are costs specified for each of the different service boxes. Furthermore, costs are defined for the use of cable and fibre. These costs increase with distance, but not necessarily linearly to reflect limitations on the length of cables and fibres.

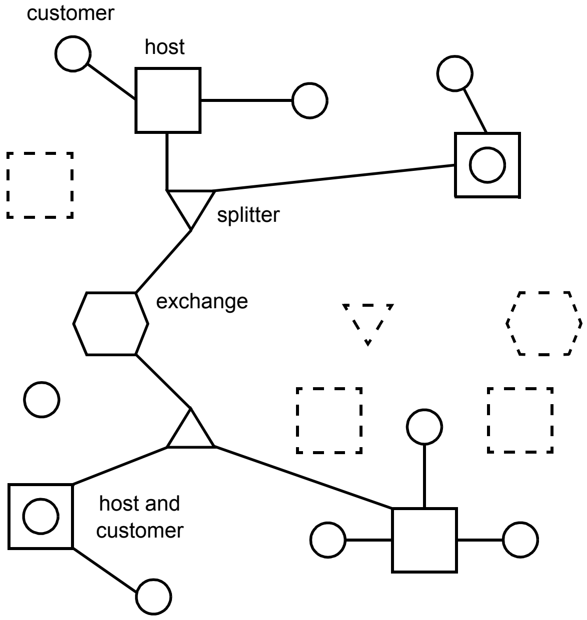

Fig 1 An example network showing the various components. It also shows (dashed) the potential locations of hosts, splitters and exchanges that have not been used. Service boxes at hosts are, however, not shown.

When a network is being designed, there are many objectives that need to be taken into account. These include the fixed cost of the network, its maintenance cost, its robustness and its extensibility. Here, we only consider two objectives. Firstly, the total cost should be as low as possible. Secondly, the unmet demand must be minimised[1]. These two objectives are weighted and summed to form a single objective that can be minimised by an ordinary EA. Although a multi-objective EA can be used to independently optimise the objective values [3], we use an ordinary EA to simplify our discussion and analysis.

3. Implementation

The EA is implemented in Java using the Eos evolutionary platform [4]. The following three sections highlight particular techniques that are used to apply the EA to the network design problem. Section 3.1 shows how the use of a neutral genetic encoding can help the EA to improve solutions. Section 3.2 shows how the EA can be combined with heuristics to good effect. Finally, section 3.3 discusses how the problem can be sub-divided to speed up optimisation.

3.1 Neutral encoding

One of the main design problems when using an EA is how to encode solutions. The encoding that is used has a significant effect on the quality of the solutions that are generated, and the time it takes to generate them. The encoding together with the genetic operators that operate on it largely determine how susceptible the EA is to sub-optimal convergence. Sub-optimal convergence occurs when an EA has converged to a solution that is not optimal but that cannot be improved through minor changes in the encoding.

[1] There is unmet demand when a host does not have sufficient service boxes to satisfy the total demand of its customers.

One mechanism to reduce the likelihood of sub-optimal convergence is the use of neutrality in the genetic encoding [5]. An encoding is neutral when changes can be made to the genetic encoding that do not affect the solution or its fitness. These neutral changes can alleviate the problem of local optima. The following example illustrates this.

Consider the situation in Fig 2(a). Here, four customers (a, b, c and d) are currently allocated to host B. However, the configuration in Fig 2(b) would actually be less expensive. Whether or not the EA is likely to find this improvement is influenced by the way solutions are encoded.

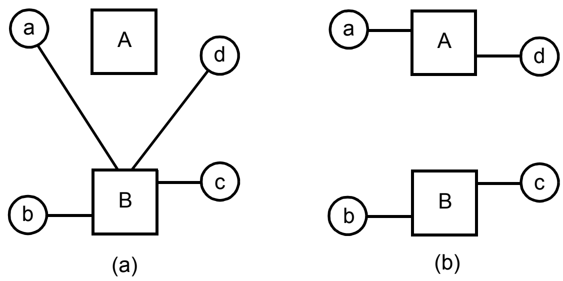

Fig 2 Two example customer-to-host allocations: (a) a sub-optimal solution, (b) the optimal solution.

If a direct encoding is used, as in Fig 3(a), it is difficult for the EA to make the improvement. The problem is that Customers a and d need to switch to Host A at the same time, to improve the solution. If only one customer switches to Host A, the solution is more expensive because the decreased cost of the cable does not offset the cost of using an extra host.

On the other hand, if a neutral encoding is used the EA is less likely to become trapped in a local optimum. Figure 3(b) shows a neutral encoding that uses grouping. Firstly, customers are mapped to one of a number of groups (in this example only two). Then, each group is assigned to a host. This allows multiple customers to switch hosts simultaneously.

In Fig 3(b), the first change is a neutral one as it does not change the solution. It does, however, set the stage for the mutation that actually improves the solution. Both mutations change only a single value, and the mutations do not have to occur simultaneously. This makes it much easier for the EA to discover the improvement.

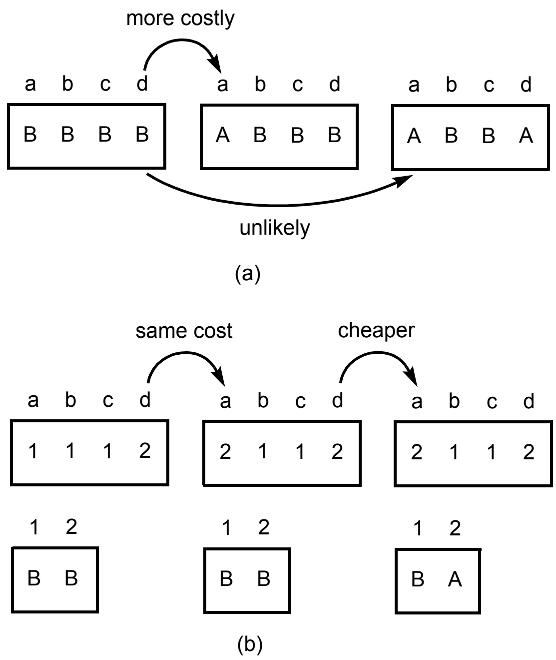

Fig 3 How the solution in Fig 2(a) can be improved for different genetic encodings: (a) a direct encoding, and (b) a neutral encoding using grouping.

We have used the above technique extensively in the genetic encoding of solutions. Customers are assigned to groups before they are allocated to hosts. Similarly, hosts are grouped before they are assigned to splitters, and splitters are grouped before they are connected to an exchange. Hosts are also grouped before service boxes are allocated.

3.2 *Hybrid evolutionary search*

An EA can be combined with a heuristic to perform a hybrid evolutionary search. There are various ways in which this can be done [6]. For the allocation of service boxes to hosts, we use a constructive heuristic that is manipulated by an EA to change the solution that it generates.

The heuristic we are using is a simple, greedy one. As long as a host still has unmet demand and spare service box capacity, the heuristic adds another service box to the host. It always selects a service box that reduces the demand. When there is more than one service box that does so, the heuristic biases its choice using preference values that are associated with each service box type. These preference values are genetically encoded so that the EA can mutate them to vary the service box allocations generated by the heuristic.

The use of a heuristic has several significant advantages. Firstly, the heuristic helps the EA to constrain the search to solutions where a host has ‘just enough’ service boxes. Without the heuristic, the EA would also have to consider solutions that are clearly inferior, i.e. those where there is still unmet demand and those with too many service boxes. Secondly, the heuristic helps the EA to simultaneously optimise the customer-to-host allocation and the service box allocation. The heuristic ensures that a change of customer for a given host is immediately reflected in the service boxes that are allocated to it. If this is not the case, it is much more difficult for an EA to find improvements. The reason is that many changes in the customer-to-host allocation, even improvements, lead to worse solutions as long as the service box allocation has not (yet) been changed to reflect this.

3.3 *Staged optimisation*

The network design problem consists of several distinct but dependent parts:

- allocation of customers to hosts,
- allocation of service boxes to hosts,
- design of the optical network, which constitutes of two parts — connecting hosts to splitters and connecting splitters to an exchange.

The EA can consider the entire problem at once, or it can separately optimise parts of the problem. The latter, staged approach has the advantage that the problem is more manageable and potentially easier to solve. The search space for each of the sub-problems is magnitudes smaller than the search space for the entire problem. However, the risk that the final solution is sub-optimal is higher. Design decisions that are optimal in the context of a sub-problem can be far from optimal when the whole problem is considered.

To investigate the extent of the trade-off, we use the following four approaches to solve the network optimisation problem:

- OneStage — solve the entire problem at once,
- TwoStageNetwork — in the first stage, allocate customers to hosts and optimise the optical network, and then, in a second stage, allocate service boxes to hosts,
- TwoStageHost — in the first stage allocate customers to hosts and optimise the service box allocation, and then, in the second stage, optimise the optical network,
- ThreeStage — in the first stage allocate customers to hosts, in the second stage, optimise the service box allocation, and then, in the third stage, optimise the optical network.

To decide when to stop an optimisation stage and switch to the next, a stage termination criterion is required. The criterion that is used is the following. A list of size H is maintained that contains a history of best objective values. After each generation of the EA the objective value of the best solution is considered. If it improves any of the objectives in the list, the list is updated accordingly. When the list has not been updated for *G* consecutive generations, the optimisation is considered to be converged, and the optimisation stage can be terminated.

4. Experiments

In this section we present two sets of experiments. In the first experiment we investigate the effect of splitting the optimisation into several stages. In the second experiment we examine the effect of the neutral encoding.

The problem to optimise is the same in both experiments. There are a hundred customers, a hundred potential hosts, ten potential splitter locations and two exchanges. All are randomly distributed in a square region of space. There are six service box types (as in Table 1). The limit on the number of service boxes is the same for all hosts and set to 10. The demand varies per customer. For each of the connection types (LAN or DS1) each customer requires between zero and twenty connections.

The configuration of the EA is the same throughout the experiments. The population size is 100, and tournament selection is used with a tournament size of 4. The recombination rate is set to 0.7. Integer genomes are recombined using 3-point crossover. The mutation operator mutates each integer value with equal probability. The probability is set such that the expected number of mutated values per genome is one. When a value is mutated, the new value is randomly chosen from the valid range. For the real-valued preference values for the service boxes, recombination and mutation operators specific to evolutionary strategies [7] are used. The termination criterion for switching between the various stages of the optimisation is the one described in section 3.3, with *H* set to 5 and *G* set to 200. For each experiment, we run the EA ten times for each configuration and average the results.

4.1 *Divide and conquer?*

Table 2 shows how each of the four variants of the algorithm performed on the basic problem. The first column shows the average running time of the EA, together with the standard deviation. The second column shows the average weighted objective value of the final solution, together with the standard deviation. The objective is defined such that a lower value corresponds to a better solution.

Table 2 Results for the first experiment, with a service box limit of 10.

	Time, sec	Best
OneStage	6171 ± 1620	425 ± 40
TwoStageNetwork	3177 ± 982	414 ± 21
TwoStageHost	5504 ± 2412	411 ± 48
ThreeStage	2930 ± 640	389 ± 16

It can be seen that the best solution is obtained by the ThreeStage algorithm. Interestingly, this is also the fastest algorithm. A possible explanation is that the problem is fairly modular so that it can be optimised effectively by considering the sub-problems separately. For the other algorithms, the search space at some of the stages is more complex and harder to navigate, thus slowing down the search and increasing the likelihood of sub-optimal convergence. These results on their own suggest that sub-dividing the problem and solving each sub-problem separately gives the best results.

Table 3, however, shows the results of an experiment very similar to that of Table 2. The only change that has been made is that in the problem the maximum number of service boxes per host has been reduced from 10 to 6.

Table 3 Results for the first experiment, with a service box limit of 6.

	Time, sec	Best
OneStage	5216 ± 1440	436 ± 13
TwoStageNetwork	1761 ± 425	520 ± 23
TwoStageHost	4802 ± 2031	451 ± 26
ThreeStage	2548 ± 597	490 ± 30

Comparing the results in Table 2 and Table 3, it can be seen that this small change in the problem had a significant effect. The best solutions are now produced by the OneStage algorithm, whereas the quality of the solutions generated by the TwoStageNetwork and ThreeStage algorithms has significantly deteriorated.

These results are easy to explain. In the first stage of each of the algorithms, customers are allocated to hosts. In both the OneStage and TwoStageHost algorithm service boxes are allocated to hosts simultaneously, whereas for the other two algorithms this happens at a later stage. What can then happen is that when customers are allocated to hosts, choices are made that turn out to be sub-optimal when the service box allocation takes place. More specifically, some hosts have so many customers assigned to them that it is impossible to satisfy the total demand given the limit on the number of service boxes. These solutions therefore inevitably incur a penalty.

What these experiments suggest is that splitting a problem up into parts can speed up the optimisation process without affecting the quality of solutions (even generating better solutions). However, care must be taken when doing so, because there is a risk that irreversible choices are made in the early optimisation stages that lead to solutions that are far from optimal.

4.2 *Invisible change*

The experiments presented so far used the neutral encoding that was described earlier. For comparison, Table 4 shows the results of an experiment where the neutral encoding has been disabled. More precisely, the grouping that was used in the encoding for the allocation of customers to hosts, hosts to splitters, and splitters to exchanges has been disabled. So here the EA independently allocates each customer to a host, etc.

Table 4 Results for the second experiment, where the neutral encoding has been disabled and the service box limit is 10.

	Time, sec	Best
OneStage	6594 ± 1452	485 ± 21
TwoStageNetwork	3413 ± 1019	441 ± 13
TwoStageHost	6873 ± 2071	494 ± 20
ThreeStage	4407 ± 1493	446 ± 15

The problem that was solved was the same as that for Table 2. Comparing the results in both tables, it can be seen that the neutral encoding gives superior results. For each of the algorithms, when neutrality is used the run time on average is shorter and the quality of the solutions is better.

5. Conclusions

We have shown how an EA can be applied to the design of passive optical networks. We demonstrated three techniques that can be applied to improve performance — using a neutral genetic encoding, using heuristic-guided evolutionary search and using staged optimisation. We presented two experiments. The first indicated the potential advantages and drawbacks of sub-dividing a problem during optimisation. The second experiment showed that the use of neutrality is indeed beneficial.

There is more work that can be done to improve the optimisation. For instance, we hope that by sub-dividing the problem more cleverly and adding more interaction between the various optimisation stages, a speed-up can be achieved while simultaneously the likelihood of sub-optimal convergence is reduced.

Acknowledgements

The exploratory work described in this paper was in response to a problem raised by Evolved Networks, a BT spin-off company. We would like to thank John Mellis for the opportunity to collaborate. Also thanks to Cefn Hoile for his suggestions about the neutral encoding to use.

References

1 Poon L F, Conway A, Wardrop G and Mellis J: 'Application of genetic algorithms to network design and planning', BT Technol J, 18, No 4, pp 32—41 (October 2000).

2 Shipman R, Botham P and Coker P: 'Coupling developmental rules and evolution to aid in planning network growth', BT Technol J, 18, No 4, pp 95—102 (October 2000).

3 Zitzler E, Deb K and Thiele L: 'Comparison of multiobjective evolutionary algorithms: empirical results', Evolutionary Computation, 8, No 2, pp 173—195 (2000).

4 Bonsma E, Shackleton M and Shipman R: 'Eos: an evolutionary and ecosystem research platform', BT Technol J, 18, No 4, pp 24—31 (October 2000).

5 Huynen M A, Stadler P F and Fontana W: 'Smoothness within ruggedness: the role of neutrality in adaptation', Proc of the National Academy of Sciences, 93, pp 397—401 (1996).

6 Ibaraki T: 'Combinations with other optimisation methods', in Bäck T, Fogel D and Michalewicz Z (Eds): 'Handbook of Evolutionary Computation', Institute of Physics, Bristol (1997).

7 Rudolf G: 'Evolution strategies', in Bäck T, Fogel D and Michalewicz Z (Eds): 'Handbook of Evolutionary Computation', Institute of Physics, Bristol (1997).

Erwin Bonsma graduated with distinction in June 1997 from the University of Twente with an MSc degree in Electrical Engineering. He subsequently specialised in non-symbolic AI at the University of Edinburgh, where he obtained an MSc degree in Artificial Intelligence with distinction in September 1998.

In early 1999, he joined BT's Future Technologies Group at Adastral Park, which is now part of the Intelligent Systems Lab. Here, he has been working on a multitude of projects, involving peer-to-peer networks, multi-agent systems, evolutionary computation and other nature-inspired techniques.

Nishan Karunatillake is a PhD student in the IAM Research Lab at the University of Southampton. He is currently reading for his PhD in 'Argumentation in Multi-Agent Interaction' under the supervision of Prof Nick Jennings.

He graduated in June 2001 with a First Class honours degree in Engineering with a specialisation in Computer Science from the University of Moratuwa, Sri Lanka. Subsequently he worked as a research student at the Intelligent Systems Lab in BT Exact from July 2001 until September 2002. Here he worked on evolutionary network optimisation and agent-based negotiation under the supervision of respectively Erwin Bonsma and Lyndon Lee.

Rob Shipman has worked in the area of nature-inspired systems for several years at Cyberlife Technology Limited and, since April 1998 until recently, in BT's Future Technologies group at Adastral Park.

Following early-life sporting preoccupations, he received a first class BEng in Electronic Systems and Control Engineering from Sheffield Hallam University, specialising in artificial intelligence, and was also awarded for exceptional performance by the Armourers and Brasiers company. In addition, he holds an MSc in Evolutionary and Adaptive Systems from the University of Sussex which was awarded with distinction. From 1994 to 1995 he studied in the Speech, Vision and Robotics group at the University of Cambridge.

He has also held several positions as a professional software engineer.

Mark Shackleton graduated from Sheffield University (UK) in 1986 with a degree in Computer Science.

He first worked for Singer Link-Miles, manufacturers of commercial flight simulators, developing real-time 3-D computer graphics algorithms and systems.

He joined the Image Processing and Computer Vision research group at BT in 1989. In this group he designed and implemented a number of systems in areas such as automatic face recognition, model-based coding, and content retrieval from images and video sequences. During this period he spent time seconded to MIT Media Laboratory working closely alongside researchers there.

In 1996 he moved across to the Future Technologies Group, now part of the Intelligent Systems Lab, BT Exact, at Adastral Park.

He is now the Future Technologies Group leader with a remit to develop novel solutions to problems using nature-inspired algorithms and approaches.

He is project manager of BT Exact's pervasive computing research programme which is seeking to address the issues of complexity inherent in the next generation of large-scale, complex, dynamic networks of computational devices.

David Mortimore began his career in the Laser Systems department of Marconi Avionics Ltd, working on advanced lasers and laser systems. He joined BT in 1982 and, after a short time with BT International writing specifications for undersea systems, he joined the Optical Communications Research Department at Adastral Park where his primary field of research was fused fibre devices. During this time he developed several novel components, such as the wavelength flattened coupler, which have been successfully licensed worldwide. In 1992 he started a new area of work concerned with the development of tools and techniques for simulating the physical layer performance of optical networks. He pioneered the development of efficient modelling methodologies for optical fibre systems, culminating in the development of a powerful PC-based optical network modelling tool, 'Fibreview'. In 1997 he joined a team working in the area of IP networks which enabled him to apply his modelling skills to the design of packet switched networks.

He left BT in 2000 to become a co-founder of Evolved Networks, a company specialising in the development of smart network planning and design tools. His current area of work, as Head of R&D, includes the development of novel algorithms and techniques related to automated planning. He is a Chartered Engineer, Chartered Physicist, Member of the IEE and IOP. He holds a BSc(Hons) degree in Applied Physics (City, London) and a PhD in Fibre Devices (Essex).