A Multi-objective Approach to Constrained Optimisation of Gas Supply Networks: The COMOGA Method

Patrick D. Surry[a,b], **Nicholas J. Radcliffe**[a,b], **Ian D. Boyd**[c]
{pds,njr} @ quadstone.co.uk; boyd @ bgers.co.uk

[a] Quadstone Ltd, 16 Chester Street, Edinburgh, EH3 7RA, UK
[b] Department of Mathematics, University of Edinburgh, King's Buildings, EH9 3JZ, UK
[c] British Gas plc, Research and Technology Division, Newcastle upon Tyne, NE99 1LH, UK

Abstract. This paper presents a new technique for handling constraints within evolutionary algorithms, and demonstrates its effectiveness on a real-world, constrained optimisation problem that arises in the design of gas-supply networks. The technique, which we call the COMOGA method (Constrained Optimisation by Multi-Objective Genetic Algorithms), borrows two crucial ideas from multi-objective optimisation and combines them with an adaptive genetic algorithm to yield a method with the same wide applicability as penalty function methods, but with significantly fewer free control parameters and the potential for greater effectiveness.

Initial results reported in this paper first compare the heuristic previously used in practice by British Gas with a genetic approach using a carefully tuned penalty function. On the problem instance studied, the genetic algorithm was able to find a feasible network with a cash cost 4% lower than the previously best-known (and installed) solution. Though successful, the penalty function method suffered from the familiar sensitivity to the settings of the parameters controlling the effective weighting of the constraint functions, and required a reverse annealing schedule to encourage feasible solutions to emerge over time.

The exercise was repeated using the COMOGA method, which treats each of the constraints—explicit or implicit—as a separate criterion in a multi-objective formulation of the problem. The same 4% improvement over the heuristic was achieved with COMOGA, in similar numbers of evaluations and with similar consistency, but with significantly less tuning. This was because of the greatly reduced number of free parameters for which values need to be selected with this method, as well as the algorithm's lower observed sensitivity to these values.

1 Introduction

It is a frequent criticism of evolutionary algorithms that published results are usually obtained with contrived problems without constraints, leading to the suggestion that evolutionary methods are unsuitable for tackling complex constrained optimisation problems. Within the community, there is a wide-spread perception that penalty function methods are a rather blunt instrument for handling general constraints (e.g. Michalewicz, 1992), exhibiting great sensitivity to the values of their many free parameters, and feeding rather too little information back to the algorithm to allow it to handle the constraints satisfactorily. While other methods are available for problems with explicit constraints (including repair methods, smart decoders and special operators incorporating problem knowledge), these do not have fully general applicability, and tend to require significant work for each new class of problems tackled. On this basis, there is an urgent need for a method that combines the generality of penalty function approaches with a greater feedback of information to the underlying search algorithm about the way in which progress is being made with the various constraints under consideration.

This study first applied a relatively straightforward genetic algorithm with a tuned penalty function to a real constrained pipe-sizing problem previously solved by British Gas. The genetic algorithm was found to produce significantly better solutions than the standard heuristics used by the company to plan the installed network, giving rise to actual cost reductions of about 4%. The COMOGA method for constraint handling (Constrained Optimisation by Multi-Objective Genetic Algorithms) was then developed. This combines two techniques previously applied in the context of multi-criterion optimisation with an adaptive selection scheme. COMOGA is shown to give similar results to those obtained using the tuned penalty function, (also beating the British Gas heuristic by 4%) but with significantly fewer free parameters and consequent experimentation.

The particular problem studied in this work is described in section 2, where the importance of constraints in the applicationis stressed. Section 3 goes on to give an overview of current approaches to constrained optimisation. Connections with multi-objective optimisation are made in section 4, and these ideas are then cast in the context of the pipe-sizing problem in section 5. The results obtained are outlined in section 6, and the main ideas are summarised in section 7.

2 Problem Specification

The design of a gas network—for example, to supply a new housing development—involves defining the layout of the network and, having done this, choosing the types of pipe to be laid. The layout is generally determined by such considerations as the routes of roads but the selection of the pipe types is tackled as a constrained optimisation problem. The important constraints on any design are that:

- the pipes selected should allow the customer demands to be met at or above a "minimum design pressure";
- each pipe (other than those incident to a source) should have at least one upstream pipe of the same or greater diameter.

Pipes are produced in a range of discrete diameters and in a number of materials, and for a given material the cost per unit length of pipe is an increasing function of diameter. The pressure drop along a pipe, for a fixed flow, is a decreasing function of diameter so larger diameters will generally give a more secure network. The problem in designing a network is therefore to select the diameters of pipes in such a way that they are large enough to provide security of supply but not larger than they need to be. The latter would amount to an over-design of the network, in that a cheaper design would have been adequate.

The cost of a network is thus the sum of the costs of the pipes it contains. However, in seeking an optimal network, the two constraints must be considered. Both the upstream pipe constraint and the minimum pressure constraint are implicit, i.e. their satisfaction can only be verified by solving the non-linear gas flow equations for the network. This is because it is only by solving the equations (or building the network!) that the upstream directions and pressures within the pipeline can be determined.

In the particular problem considered, the network contained 25 pipes, each of which could be selected from six possible sizes, giving rise to a search space of size $6^{25} \simeq 3 \times 10^{19}$. The pipes connect 25 nodes, 23 of which are (varying) demand nodes and two of which are pressure-defined source nodes. (Flow-defined source nodes may also be specified.) The network is a real one, which was actually built, with pipe sizes determined using the heuristic method discussed in section 6.2.

The density of valid networks (ones which satisfy the constraints) in the search space is extremely low—random sampling of more than 3×10^7 points produced only a single admissible configuration.

3 Constrained Optimisation

Many other optimisation problems can also be phrased as

Minimise (maximise)

$$f : \mathcal{S} \longrightarrow \mathbb{R}$$

subject to [the solution $x \in \mathcal{S}$ satisfying certain equalities or inequalities].

The equations or inequalities that the solution x must satisfy are known as constraints. Indeed, in some applications, the objective function f is discarded with the goal being simply to find some solution that satisfies the set of constraints (so-called constraint satisfaction problems).

The constraints on x are conveniently divided into two categories—implicit and explicit. Explicit constraints are those which can be reduced to simple, explicit conditions on x, while implicit constraints are those which specify a condition on some (possibly complex) function of x.

When a constrained problem is tackled using an evolutionary algorithm, three main approaches are available. The first is to build and use genetic operators that "understand" the constraints, in the sense that they never produce infeasible solutions (ones that violate the constraints). These are often called *greedy decoders*. The search is thus reformulated as an unconstrained optimisation problem over the reduced space $\mathcal{S}_f$ (the *feasible* region), which is illustrated in figure 1. The diagram shows the image of the search space under the vector-valued function $\mathcal{I} : \mathcal{S} \longrightarrow (\mathbb{R}^+)^2 \times \mathbb{R}$ where $\mathcal{I} \triangleq (c_1, c_2, f)$ with $c_i(x)$ measuring the degree to which x violates the ith constraint and $f(x)$ giving its cost (to be minimised). This approach is advocated by Michalewicz (1992), and Radcliffe (1994). Michalewicz has shown how genetic operators can be built that "understand" linear constraints in this sense.

A second approach is to use repair mechanisms to produce feasible solutions from infeasible ones (mapping $\mathcal{S} \backslash \mathcal{S}_f \longrightarrow \mathcal{S}_f$). When such mechanisms are employed, a further

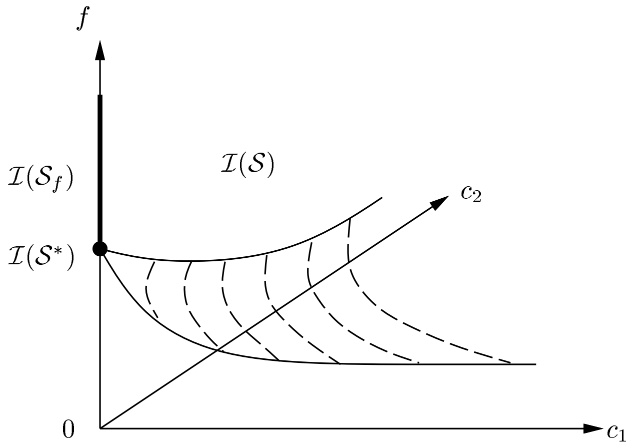

Fig. 1. A constrained optimisation problem to minimise cost f while satisfying two implicit constraints is illustrated. Points in the search space $\mathcal{S}$ are shown under the three-dimensional mapping $\mathcal{I} \triangleq (c_1, c_2, f)$. The c_1 and c_2 axes measure the degree of constraint violation, so that points in the feasible region, $\mathcal{S}_f$, are mapped to the line where both are zero. The desired set of optima $\mathcal{S}^*$ is the set of feasible points with minimum cost. All solutions in $\mathcal{S}$ are mapped to points on and above the shaded surface.

choice is available, namely whether to write the repaired solution back to the genome, or to use it during evaluation, but then to leave the (infeasible) genome intact. The former approach, which is known as Lamarckism, has the advantage of generally allowing faster local improvement, but can make it harder for the search to traverse infeasible regions of the search space, particularly when $\mathcal{S}_f$ is strongly disconnected with respect to the genetic operators. The latter approach, (which has some parallels with the Baldwin effect) has converse advantages and disadvantages. Davis & Orvosh (1993) present anecdotal empirical evidence that writing the repaired solution back to the genome probabilistically, about 5% of the time, is the best option.

In a problem with implicit constraints, it is often at least as difficult to determine whether a solution is feasible as to evaluate its cost. In such a case, neither of these first two techniques is likely to be appropriate. The final approach, which is generally accepted to be the least attractive, but most universally applicable, is to employ a penalty function. Here, the search is treated as an unconstrained problem over $\mathcal{S}$, but the objective function is modified for infeasible solutions by adding terms which degrade their performance. In general, the size of penalty added reflects in some way the *degree* of constraint violation. It is also reasonably standard practice (e.g. Richardson *et al.,* 1989; Michalewicz, 1992) to increase the size of penalties during the course of a run (reverse annealing), so that while a degree of violation is tolerated early in early generations, this toleration reduces over time. This ensures that, after sufficient time, the optimal solutions to the unconstrained problem using the modified objective function coincide with the optimal solutions to the original constrained problem.

Although penalty functions are more-or-less universally applicable, they exhibit a number of drawbacks. First, they are weak, in the (formal) sense that they do not provide any problem-specific information to the algorithm. This contrasts with repair mechanisms and problem-specific move operators that exploit understanding of the constraints to provide stronger guidance to the algorithm, but such techniques are not applicable for general constraints. Secondly, the choice of weighting for the constraints is a somewhat subtle matter, particularly when there are many, and increases yet further the number of free parameters to the evolutionary algorithm. The resulting quality of solution obtained—in fact, the likelihood of finding *any* feasible solution—may be extremely sensitive to the values chosen. In the next section, we propose a method which avoids this difficulty, by appealing to the methods of multi-criterion optimisation.

4 Links to Multi-Criterion Optimisation

In many real-world optimisation problems there is not a single objective but a set of criteria against which a solution may be measured. Such problems are often known as *multi-objective* or *multi-criterion* optimisation problems, and are defined by a set of objective functions $f_1, f_2, \ldots, f_N$ over the search space $\mathcal{S}$, each of which should ideally be minimised (maximised).

Perhaps the most common approach to multi-criterion optimisation is to form a new objective function F that is a weighted sum of the individual objectives,

$$F = \sum_{i=1}^{N} \alpha_i f_i, \quad \alpha_i \in \mathbb{R}^+$$

and to seek to minimise this sum.

If there exists a solution $x^* \in \mathcal{S}$ that simultaneously succeeds in minimising each of the f_i, this approach can be reasonably satisfactory, because in this case, successful optimisation of F will also optimise each f_i. In the more general case, however, the component objectives f_i will *compete,* in the sense that improvement against one will in some cases require a degradation against another. In this case, the approach of forming a weighted sum is less attractive, because the choice of weights α_i will determine the trade-off between the various component objectives that optima of the combined function F will exhibit. This is particularly unsatisfactory in cases where the various objectives are *non-commensurate,* in the sense that trade-offs between them are either arbitrary or meaningless. A good example of this might arise when seeking to maximise profit while minimising ecological damage, where most people would accept that any assignation of economic cost to ecological damage is at best arbitrary.

In the case of multi-objective problems with competing, non-commensurate criteria, a more satisfactory approach is to search not for a single solution but for that set of solutions that represent the "best possible trade-offs." Such solutions are said to be *pareto-optimal,* (after Vilfredo Pareto who first advanced the concept) and are characterised by introducing the notion of domination. A solution x is said to *dominate* another solution y if its performance against each of the objective functions is at least as good as that of y, and its performance is better against at least one objective, i.e. if and only if

$$\begin{aligned} \forall i \in \{1, 2, \ldots, n\}: &\quad f_i(x) \leq f_i(y) \\ \text{and } \exists\, j \in \{1, 2, \ldots, n\}: &\quad f_j(x) < f_j(y). \end{aligned}$$

Clearly in this case, x may reasonably be said to be a superior solution to y. The *pareto-optimal set* (or *front*) $\mathcal{P}$ is the set of solutions that are not dominated by any other solution in the search space, i.e.

$$\mathcal{P} \triangleq \left\{ x \in \mathcal{S} \;\middle|\; \nexists\, y \in \mathcal{S} :\; y \text{ dominates } x \right\}.$$

It is natural in population-based search algorithms such as those normally used in evolutionary computing to consider trying to use the population to hold solutions that represent different trade-offs. Relatively simple modifications to the selection (and perhaps the replacement) method may be all that is required to effect this. A number of schemes have been proposed, most of which are based around the notion of only allowing selective advantage between solutions when one dominates another. Fonseca & Fleming (1994) provide an overview of many such techniques. The effectiveness of these methods is further enhanced when combined with some form of niching, to encourage greater diversity in the population. Niching methods include structured population models (e.g. Norman, 1988,Gorges-Schleuter, 1989, Manderick & Spiessens, 1989), sharing (Goldberg & Richardson, 1987), and crowding (Cavicchio, 1970; DeJong (1975)).

It is clear that the constraint satisfaction problem is equivalent to the simple class of multi-objective problems discussed above, in which all objectives can be minimised simultaneously. (Consider again figure 1, where we are now only required to find a solution in $\mathcal{S}_f$. The dotted lines, if extended upwards as vertical manifolds, might indicate a series of progressively dominating surfaces, converging on $\mathcal{I}(\mathcal{S}_f)$—the pareto-optimal set in this case.) Although, in such a case, minimising a penalty function expressing the degree of constraint violation would be the most common approach, we suggest that a more appropriate strategy when using evolutionary techniques is to exploit the ability of the population to hold many different possible trade-offs, along with the simple techniques for general multi-criterion optimisation discussed above, to let the algorithm dynamically "discover" an appropriate trajectory by which to approach the feasible region.

This idea can then be extended to the constrained optimisation problem, where not only must several constraints be satisfied, but a given objective function f must also be minimised. Here we think of f as an extra criterion which is of lesser importance than any of the "constraint criteria", i.e. there is no acceptable trade-off between minimising (satisfying) the constraints, and minimising f. This motivates the COMOGA approach, which is introduced in section 5.2.

5 Approaches to the Pipe-sizing Problem

The pipe-sizing problem introduced in section 2 is a typical example of the constrained optimisation problem discussed in section 3. Here we have a simple objective function (the cost of the pipes in the network), and two implicit constraints which must be satisfied. We will first look at the conventional penalty function approach, in section 5.1, and then at the COMOGA approach, which is motivated by the ideas from multi-criterion optimisation discussed above.

5.1 Penalty Function

Because of the implicit nature of the constraints in the pipe-sizing problem, the only applicable conventional approach is to use a penalty function, as it would be extremely difficult to construct genetic operators that respected them, and prohibitively expensive to use a repair mechanism.

The form of the modified objective function used (as suggested by work such as Richardson *et al.,* 1989; Michalewicz, 1992) was

$$cost(D_j) = \sum_{j=1}^{N_{pip}} l_j \cdot c(D_j) + \alpha n_{gen}^{k_1} \cdot (p_{des} - p_{min})^{k_2} + \beta n_{gen}^{k_3} \cdot \sum_j (D_j - D_k)^{k_4}$$

where the first term is the cost of the pipes as a function of their diameters and (fixed) lengths, the second term is the minimum pressure constraint (penalising networks with pressures less than some specified design value), and the final term is the upstream pipe constraint with summation over pipes j where there is no upstream pipe of greater or equal diameter and D_k is the diameter of the largest upstream pipe from pipe j. Note that this involves no fewer than six control parameters (the four k_i, α, and β) for two constraints.

Values were selected for the constants α and β that normalised nominal values of the penalties to the same scale as the basic cost of the network. The various exponents were selected in order to make the penalties grow at roughly the same rate as networks became "worse" at satisfying the constraints (values of $k_2 = 0.5$ and $k_4 = 1.0$ were used). The annealing parameters k_1 and k_3 were subject to some experimentation, but 0.2 was found to be an effective value for both.

5.2 The COMOGA Approach

In section 4, we suggested that the methods of evolutionary multi-objective optimisation can be applied directly to the constraint satisfaction problem. However, the situation is complicated somewhat by the additional requirement of minimising some function over the feasible region.

We can conceptually label all members of the search space $\mathcal{S}$ with some measure of their pareto ranking based on constraint violation, either by conceptually peeling off successive non-dominating layers (Goldberg, 1989), or by assigning to each solution a "rank" equal to the number of solutions which dominate it (Fonseca & Fleming, 1993). In practice, we must be satisfied with calculating rankings based on the current population of solutions, rather than the full search space, but the principle is the same. We denote this ranking function $R : (\mathbb{R}^+)^N \longrightarrow \mathbb{Z}^+$, where N is the number of constraints. Since every solution has some cost value associated with it, we can form the two-dimensional mapping $\mathcal{I}_R : \mathcal{S} \longrightarrow \mathbb{Z}^+ \times \mathbb{R}$, with $\mathcal{I}_R \triangleq (R \circ (c_1, ..., c_N), f)$, leaving us with the two-objective problem illustrated in figure 2. However, we desire not simply solutions on the pareto-optimal surface $\mathcal{P}$, but rather solutions in the intersection of the pareto-optimal set with the feasible region (as constraint satisfaction is "more important" than cost minimisation, assuming that the constraints are not "soft").

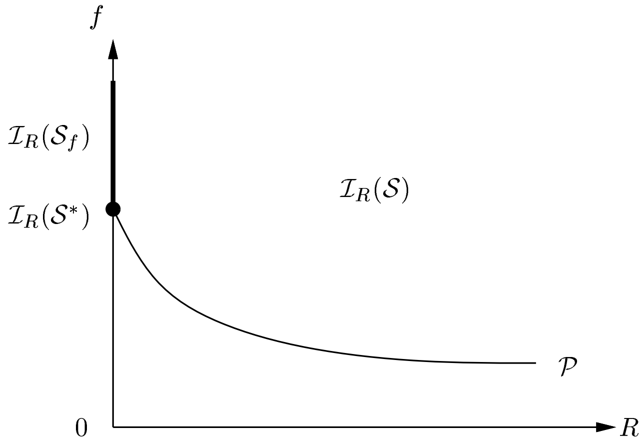

Fig. 2. The constrained optimisation problem with N constraints can be re-cast as a two-objective problem by assigning a pareto rank based on constraint violation. The pareto-optimal set $\mathcal{P}$ is the set of non-dominated solutions under $\mathcal{I}_R \triangleq (R \circ (c_1, ..., c_N), f)$. The feasible set is mapped to the line segment $\mathcal{I}_R(\mathcal{S}_f)$, and the desired set of optima is mapped to their intersection, $\mathcal{I}_R(\mathcal{S}^*)$. The search space $\mathcal{S}$ is mapped to points on and above $\mathcal{P}$.

One possible approach is to use a sub-ranking scheme, where only solutions with equal pareto rank for constraints are distinguished on the basis of cost. However, this seems likely to result in an evolutionary process which first concentrates on the constraint satisfaction problem (hence sampling points in the feasible region essentially at random) and only once this is solved tries to reduce cost. This "approach from above" not only lacks the desirable property of being able to combine low-cost, nearly-feasible solutions with higher-cost feasible ones, but may be an extremely poor way to search $\mathcal{S}_f$ if it is a highly sparse and disconnected subset of $\mathcal{S}$.

An appealing alternative approach is to enlist the ideas of Schaffer (1985). In his *vector evaluated genetic algorithm* (VEGA), he selects $1/N$ of the population based on each of the N objective functions. Although his use of proportional selection is criticised for its tendency to favour the development of "specialist" populations that excel in one objective function (Richardson *et al.,* 1989), use of rank-based (or equivalently tournament) selection may help to alleviate this (Fonseca & Fleming, 1994).

The suggestion in our case is to use binary tournament selection (Goldberg, 1989), choosing, with probability p_{cost}, cost f as the tournament criterion, while using the constraint rankings R of the solutions the remainder of the time. In cases where the selected attributes are equal, the other attribute is compared. Any fixed value of p_{cost} will induce an overall probability of reproduction equal to some linear combination of the reproductive probabilities with respect to the two attributes, with *population-dependent* weights (Fonseca & Fleming, 1994). Although such a fixed p_{cost} may favour convergence to some non-feasible point on the pareto-optimal curve, it is clear that as $p_{cost} \to 0$, the process increasingly favours constraint rank until in the limit of $p_{cost} = 0$ we are essentially solving the constraint-satisfaction problem; seeking feasible solutions regardless of cost (unless the constraint rankings are equal—this is equivalent to the sub-ranking approach described above). We thus hope that some intermediate non-zero value will allow us to find feasible solutions of low cost. This is illustrated in figure 3.

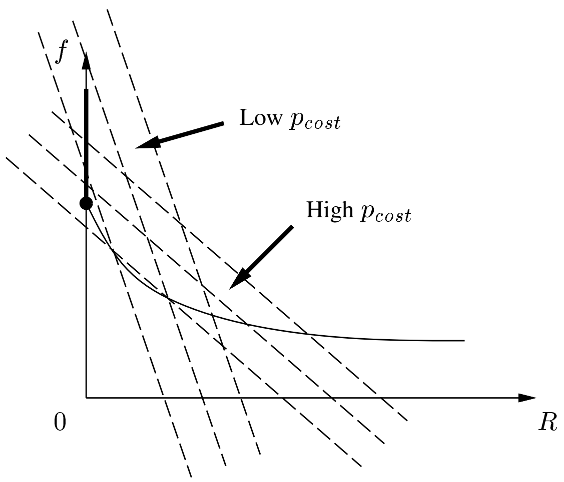

Fig. 3. Using a VEGA-like scheme of selecting probabilistically with respect to one of the two objectives (cost or constraint rank), we induce a perceived fitness of some population-dependent weighted combination of the objective values. As p_{cost} tends to zero, the scheme favours constraint rank more, and cost less.

To avoid the problem of fixing a particular value for p_{cost}, we propose to change the value adaptively by setting a target proportion τ of feasible solutions in the population. (τ is similar to the flip threshold of Schoenauer & Xanthakis, 1993.) We start by choosing some arbitrary value for p_{cost}, say 0.5, and some desired proportion of feasible solutions, e.g. $\tau = 0.1$. After each generation, if some moving average over the last several generations of the number of feasible solutions in the population is not close to τ, and if the trend in the average is not towards τ then we adjust p_{cost} up or down accordingly: if the actual proportion is too low, we decrease p_{cost}, e.g. $p_{cost} \leftarrow (1-\varepsilon)p_{cost}$, and conversely, if the proportion is too high, we increase it, e.g. $p_{cost} \leftarrow 1-(1-p_{cost})(1-\varepsilon)$. This does, of course, introduce several new parameters to the algorithm, which we were trying to avoid, but we find in practice that the scheme is remarkably robust to them, in contrast to typical penalty function parameters. This leads to the COMOGA method, which is outlined below.

The COMOGA Method

1. Calculate constraint violations for all solutions.
2. Pareto rank based on constraint violations (e.g. by counting the number of members of the population dominated by each solution).
3. Evaluate the cost (fitness) of solutions.
4. Select an (expected) proportion p_{cost} of parents based on cost, and the others based on constraint ranking.
5. Apply the genetic move operators (recombination, mutation etc.)
6. Adjust p_{cost} if the proportion of feasible solutions in the population is not close to the target proportion, τ. Lowering p_{cost} favours feasible solutions and raising it favours lower cost solutions.

The COMOGA scheme has several beneficial features. First, and foremost, it removes the necessity for the many parameters of a penalty function which must be determined empirically. Secondly, it exploits VEGA's tendency to favour extreme solutions, as in this case we are only ultimately interested in solutions which excel at constraint satisfaction (have low constraint rank). Thirdly, the adaptive approach to specifying p_{cost} allows the algorithm to find its own trajectory to approach the desired optimal values.

6 Empirical Framework and Results

6.1 Genetic Representation and Operators

The representation used for the pipe sizes in the network is a variable cardinality integer string. A genome is a sequence of n integers $a_1a_2\cdots a_n$ with $a_i \in \{0,1,\ldots,C_i-1\}$, where C_i is the cardinality (number of alleles) of the ith gene and n is the number of pipes in the network. (The particular problem instance tackled happened to have $C_i = 6$ for each of the 25 pipes but this is not generally the case.) This representation was built as a generic library for the *Reproductive Plan Language*, RPL2 (Surry & Radcliffe, 1994, Radcliffe & Surry, 1994).

We used random mutation with rate 0.025 (we allow an allele to be replaced by the same value), combined with non-cyclic creep mutation (see Davis, 1991) with rate 0.05. Creep mutation is more appropriate (and effective) in the context of the pipe-sizing problem since the alleles in this case correspond to an ordered list of pipe sizes, so that consecutive integers represent similar pipe sizes.

Initial experiments made use of N-point crossover, with $1 \le N \le 5$, but this was found to be less effective than parameterised uniform crossover with suitable bias

(Spears & DeJong, 1991; Syswerda, 1989). We used a bias of 0.6, and a crossover rate of 1.0. Although uniform crossover is cited as weaker than N-point crossover in preserving short schemata, this is relevant only if there is greater than average correlation between adjacent genes in the genome (strong linkage). In the case of the pipe-sizing problem, it is difficult in general to define an ordering in which this is necessarily the case, making the uniform crossover operator appropriate for the problem. It is possible that a labelling derived by minimising the bandwidth of the network's connectivity matrix would increase the effectiveness of N-point crossover for the pipe-sizing problem, but this has not yet been explored. A further alternative would be to use an N-cut crossover based on the fixed-topology graph making up the network.

Elitism was used to preserve the best member of the population, and any duplicates that were created were discarded rather than include multiple copies of them in the population.

All results presented here used an unstructured ("panmictic") population. Although some experimentation with various forms of fine-grained and island structures was carried out with the original penalty-function approach, they did not yield significant benefits on this problem.

6.2 Heuristic Approach

The current heuristic technique used by British Gas was applied to the problem, in order to compare its performance to the genetic approach. In fact the network had been installed using the results obtained from the heuristic.

The heuristic determines a good configuration of pipe sizes by first assuming a constant pressure drop over the whole network and guessing some initial pipe sizes. This always yields a valid network (i.e. one that satisfies the constraints), but does not normally produce an optimal configuration. The heuristic proceeds by locally optimising this solution, repeatedly trying to reduce single pipe diameters while maintaining a valid network. Eventually this process terminates when no pipe size can be reduced while maintaining network validity.

The algorithm takes on the order of ten seconds on a 486 PC (25MHz) to reach its best configuration for the problem under study. A schematic (which does not represent differences in pipe lengths and source/demand requirements) of this solution appears in the left part of figure 4.

6.3 The Genetic Algorithm with a Penalty Function

A wide range of penalty function parameters were tested before arriving at the values reported in section 5.1. As has been widely reported previously, the quality of the resulting algorithms is highly sensitive to these values, with small changes often resulting in non-converging runs.

A steady-state update scheme was used, with fitnesses re-calculated once every generation, since the penalty function is dependent on the current generation number. Binary tournament selection and replacement with parameter 1.0 were found to be the best update strategy. It is interesting to note that the algorithm was also quite sensitive to some of these other parameters, not involved in the penalty function—for instance, a significant degradation of performance was observed as the binary-tournament parameter was changed. Population sizes from 50 to 500 were tested, with the most efficient being about 100 individuals. Runs typically involved ten-to-twenty thousand evaluations, with a stopping condition of five thousand evaluations with no improvement.

The technique (with good parameters) produced consistently good results, although it did not always converge to the same optimal solution. In most cases it found networks which were better, often significantly so, than that determined by the heuristic approach. In almost all cases the algorithm found a valid network by the end of the run (i.e. one in which the penalty terms were zero). Run times for typical populations of 100 networks through 100 generations were of the order of several minutes on a Sun SPARC 2 workstation. The best result was a network which was approximately 4% cheaper than the heuristic solution. A schematic of this network is shown in figure 4.

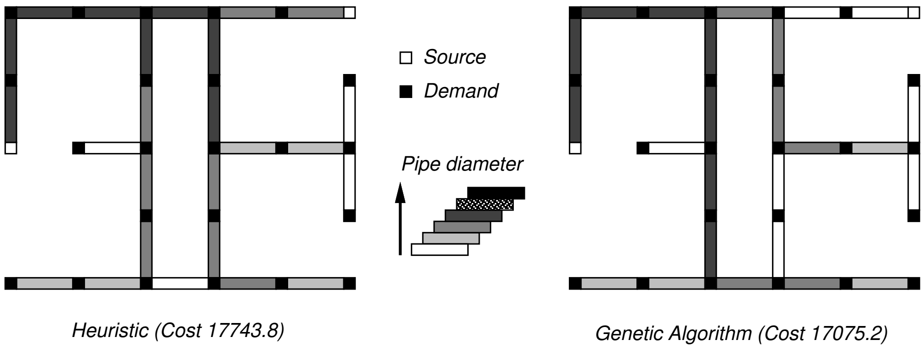

Fig. 4. The genetic algorithm finds a solution approximately 4% better than that found by the heuristic technique. The networks are shown only schematically, so that the pipes are actually of different lengths. The demand and supply requirements are also different at each node. Both networks are valid in that they satisfy both the upstream-pipe and minimum pressure constraints. The genetic algorithm result is typical of a panmictic population of population size 100, running for 100 generations. Run time for the heuristic is a few seconds, as compared to a few minutes for the genetic algorithm.

6.4 The COMOGA Method

The COMOGA method, outlined in section 5.2 was implemented using the ranking technique described by Fonseca & Fleming (1993). Each member of the initial population was assigned a rank according to constraint violation by counting the number of members in the population by which it was dominated. Binary tournament selection with parameter 1.0 was used (with tournaments based on cost value with probability p_{cost}, and otherwise on constraint ranking) to select parents, and the resulting child was reinserted using a replace-worst (with worst being with respect to cost with probability

p_{cost}) scheme. The nature of the ranking scheme means it is easy to subtract the effect of the deleted individual and add the effect of the new individual without re-ranking the entire population. Tournament replacement was also investigated, but the more aggressive replace-worst strategy proved superior.

As with the penalty function approach, a variety of population sizes, mutation and crossover rates, and so forth were investigated. Results were best with populations of about 100 individuals, and with the same stopping conditions, runs lasted for similar numbers of evaluations, and produced similar quality solutions (the same "best" solution from the penalty-function approach was found consistently). In general, algorithm performance was much less sensitive to the control parameters than were the penalty function experiments. Most importantly, the COMOGA scheme was not particularly sensitive to the method used for adapting the p_{cost} parameter, nor to the target proportion of feasible solutions, τ.

As noted in the introductory sections, the overall performance of the algorithm employing COMOGA was very similar to that of the best penalty function approach found, both in terms of computational effort required and frequency of finding the best solutions. However, significantly less experimentation was required to find values for COMOGA's parameters that work well than was the case with the penalty function method.

7 Summary

A new approach of general applicability to constrained optimisation—the COMOGA method—has been introduced. This technique treats each constraint as a separate criterion, then uses a form of pareto ranking to order solutions in terms of their constraint violation, and finally employs a self-adaptive form of Schaffer's VEGA scheme, using cost as one criterion and overall ranked performance against constraints as another.

COMOGA has been successfully used to tackle a real-world, gas-network pipe-sizing problem with implicit constraints, and shown to produce solutions of similar quality, with similar frequency, as a genetic algorithm using a traditional penalty function approach with a tuned penalty function. Both forms of genetic algorithm were able to reduce the cash cost by 4% over the existing heuristic technique used to design the network previously installed.

The COMOGA method uses the memory implicit in the population to "discover for itself" the relative worth of different achievable combinations of constraints and objectives. The population thus forms not just a pool of good solutions among which recombination takes place, but a context in which to determine the fitness of any one member—the effective weighting of the various constraints is determined by the population, as is the relative weighting of constraint satisfaction and cost minimisation. This contrasts with a penalty function approach, where both are determined *a priori,* and appears to carry the singificant benefit of reducing the sensitivity of the genetic algorithm to the values of the free parameters.

While it remains to be seen whether the COMOGA approach will be as successful for other constrained optimisation problems, the preliminary results are very promising.

Acknowledgements

The initial work in this study was carried out as a collaborative project between Edinburgh Parallel Computing Centre and British Gas plc. Patrick Surry and Nicholas Radcliffe completed the study after leaving EPCC to join Quadstone Ltd and to take up positions as research student and visiting professor respectively in the Department of Mathematics at the University of Edinburgh.

References

D. J. Cavicchio, 1970. *Adaptive Search Using Simulated Evolution*. PhD thesis, Univerity of Michigan.

Lawrence Davis and David Orvosh, 1993. Shall we repair? Genetic algorithms, combinatorial optimization, and feasibility constraints. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo).

Lawrence Davis, 1991. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold (New York).

Kenneth A. De Jong, 1975. *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan.

Carlos M. Fonseca and Peter J. Fleming, 1993. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423. Morgan Kaufmann (San Mateo).

Carlos M. Fonseca and Peter J. Fleming, 1994. Multiobjective evolutionary algorithms: An overview. In Terence C. Fogarty, editor, *AISB Workshop on Evolutionary Computing, University of Leeds*.

David E. Goldberg and Jon Richardson, 1987. Genetic algorithms for multimodal function optimisation. In *Proceedings of the Second International Conference on Genetic Algorithms*. Lawrence Erlbaum Associates (Hillsdale).

David E. Goldberg, 1989. *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley (Reading, Mass).

Martina Gorges-Schleuter, 1989. ASPARAGOS: an asynchronous parallel genetic optimization strategy. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 422–427. Morgan Kaufmann (San Mateo).

B. Manderick and P. Spiessens, 1989. Fine-grained parallel genetic algorithms. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 428–433, San Mateo. Morgan Kaufmann Publishers.

Zbigniew Michalewicz, 1992. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag (Berlin).

Michael Norman, 1988. A genetic approach to topology optimisation for multiprocessor architectures. Technical report, University of Edinburgh.

Nicholas J. Radcliffe and Patrick D. Surry, 1994. The Reproductive Plan Language RPL2: Motivation, architecture and applications. In J. Stender, E. Hillebrand, and J. Kingdon, editors, *Genetic Algorithms in Optimisation, Simulation and Modelling*. IOS Press (Amsterdam).

Nicholas J. Radcliffe, 1994. The algebra of genetic algorithms. *Annals of Maths and Artificial Intelligence*, 10:339–384.

John T. Richardson, Mark R. Palmer, Gunar Liepins, and Mike Hilliard, 1989. Some guidelines for genetic algorithms with penalty functions. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 191–195. Morgan Kaufmann (San Mateo).

J. David Schaffer, 1985. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of an International Conference on Genetic Algorithms*. Lawrence Erlbaum.

Marc Schoenauer and Spyros Xanthakis, 1993. Constrained GA optimisation. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo, CA).

William M. Spears and Kenneth A. De Jong, 1991. On the virtues of parameterised uniform crossover. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 230–236. Morgan Kaufmann (San Mateo).

Patrick D. Surry and Nicholas J. Radcliffe, 1994. RPL2: A language and parallel framework for evolutionary computing. In Y. Davidor, H.-P. Schwefel, and R. Männer, editors, *Parallel Problem Solving from Nature III*, pages 628–637.

Gilbert Syswerda, 1989. Uniform crossover in genetic algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann (San Mateo).