
Designing Crushers with a Multi-Objective Evolutionary Algorithm

L. Barone

Department of Computer Science & Software Engineering
The University of Western Australia
Nedlands, WA, Australia
luigi@cs.uwa.edu.au;lyndon@cs.uwa.edu.au

L. While

P. Hingston

School of Computer and Information Science
Edith Cowan University
Mt Lawley, WA, Australia
p.hingston@ecu.edu.au

Abstract

This paper describes the use of a multi-objective evolutionary algorithm to solve an engineering design problem - determining the geometry and operating settings for a crusher in a comminution circuit for ore processing. The outcome is a tool for consulting engineers that can be used to create and explore candidate designs for various scenarios. The tool has proved capable of deriving designs that are clearly superior to existing designs, promising significant financial benefits. The approach is flexible enough to be applied to a variety of similar problems.

1 INTRODUCTION

Evolutionary algorithms are increasingly finding applications in engineering design tasks. In this paper we describe a study, supported by Rio Tinto Ltd, which uses evolutionary algorithms to optimise the performance of a comminution circuit for iron ore processing. In work reported earlier, (Hingston, 2002), a simple evolution strategy algorithm was used to solve this problem. We have restated the details of the problem description here for completeness. In the present study, we report on a further development - a multi-objective algorithm - and compare the two approaches.

The performance of a processing plant has a large impact on the profitability of a mining operation, and yet plant design decisions are often guided more by engineering intuition and previous experience than by analysis. This is because plants are extremely complex to model, so engineers often must rely on simulation tools to evaluate and compare alternative hand-crafted designs. This is a time-consuming process and the lack of an analytical model means that there is little theoretical guidance to narrow the search for better solutions. Evolutionary algorithms can be of great benefit here, providing a means to search large design spaces and present the

engineer with superior designs optimised for different operating scenarios.

In order to test the applicability of evolutionary algorithms in this setting, a representative problem was chosen by Rio Tinto. The task was to find combinations of design variables (including geometric shapes and machine settings) to maximise the capacity of a simple comminution circuit, whilst also minimising the size of the product. Earlier work in (Hingston, 2002) showed the effectiveness of a single-objective evolution strategy algorithm for this task. However, the multi-objective approach described in this paper offers clear advantages over the single-objective algorithm.

We begin the paper with a description of the problem, including a brief background on crushers and comminution circuits. Section 3 describes our mapping of the problem to an evolutionary algorithm, including the genetic representation, genetic operators and selection methods. Section 4 presents some illustrative results. Finally, we discuss future enhancements to the system and plans to extend the work to include greater complexity in the simulation model, including circuits.

2 BACKGROUND

Crushing and grinding of rocks and other particles has many important applications, including coarse crushing mined ore and quarry rock, fine grinding of coal for power station boilers, and for production of paint, ceramics, cement and other materials. It has been estimated that several billion tons of material is crushed and ground annually (Hiorns, 1971). Thus optimisation of crushing operations offers large potential economic benefits. For example, in the area of energy savings, Napier-Munn et al ((Napier-Munn, 1996), p1) quote a report of the U.S. National Materials Advisory Board in 1981, which estimated that realistic improvements in crushing-related activities could result in energy savings of more than 20 billion kWh per annum. Other benefits of optimisation of crushing and grinding in mineral processing operations include reduced operating costs, increased throughput and thus value production, and improved downstream performance.

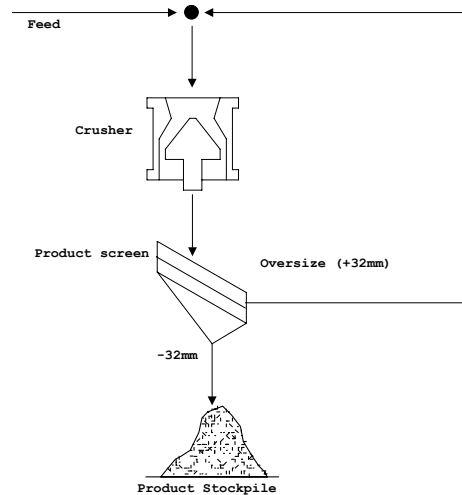


Figure 1 - The simple circuit used in this study

2.1 CRUSHERS AND CIRCUITS

In this section, we provide a brief background on crushers and how they are used in comminution circuits. The interested reader could consult, for example, (Napier-Munn, 1996) for more detailed information. “Comminution” refers to the collection of physical processes that can be applied to a stream of ore to change the size of the particles in the stream. Examples include crushing and grinding (which break ore particles into smaller particles), and screening (which separates ore into several streams of different particle sizes). The purpose of comminution is to transform raw ore into a more usable or more saleable product or to prepare it for further processing. A “comminution circuit” consists of a collection of processing units (crushers, screens etc) connected together (by conveyor belts, for example), possibly containing loops (hence the use of the word “circuit”). One or more streams of ore (the “feed”) enter the circuit and one or more streams of transformed material (the “product”) exit the circuit.

Figure 1 shows the simple circuit that was used in this study. The feed comes in on a conveyor from the top left and enters the crusher. The crushed ore is then passed through a screen that allows particles less than 32 mm to pass through and report to product. Particles larger than this (the “oversize”) are recycled back to the crusher. Thus the input to the crusher is a combination of feed and recirculating oversize.

The type of crusher used here is a “cone” crusher. Figure 2 is a schematic diagram of a typical cone crusher. Material is introduced into the crusher from above, and is crushed as it flows downwards through the machine. The inner crushing surface, or “mantle”, is mounted on the

conical crushing head and is driven in an eccentric motion swivelling around the axis of the machine. The outer crushing surface, or “bowl”, is held stationary. Material flows into the crushing chamber from above, and is crushed between the two surfaces by compressive forces due to the eccentric motion. After compression, the chamber widens and allows material to flow to lower parts of the crushing chamber, and eventually to fall through and exit the machine.

The gap between the bowl and the crushing head at the closest point in the cycle is called the “closed-side setting”. This can be reduced to obtain a narrower chamber and finer crushing. The two crushing surfaces are covered by replaceable steel liners (shaded in Figure 2), which can be manufactured with different cross-sectional shapes. The eccentric angle and speed of revolution of the head can also be adjusted. These variables contribute to the performance characteristics of the crusher.

2.2 SIMULATING CRUSHERS

Fitness is evaluated using a simulation of a single cone crusher. The inputs to the simulation are the:

- Physical properties of the feed (composition, hardness etc);
- Size distribution of the feed (the proportion of particles in different size fractions);
- Geometry of the mantle and bowl liners;
- Closed-side setting;
- Rotational speed of the head; and
- Eccentric angle of the head.

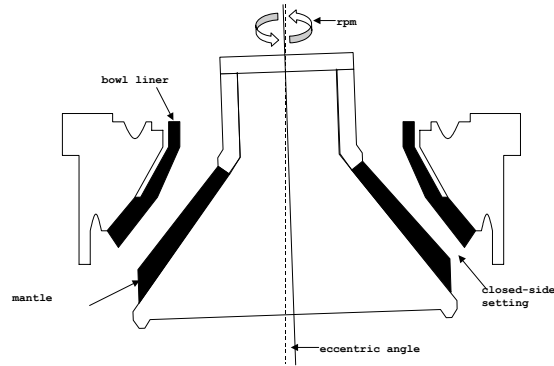


Figure 2 - Schematic diagram of a cone crusher (after (Napier-Munn, 1996) Figure 6.3)

The final four of these were selected as the design variables for the chosen problem. The outputs of the simulation are the:

- Size distribution of the product;
- Power needed to crush the feed; and
- Maximum amount of material that can flow through the crusher without overloading the crusher (its “capacity”).

From these outputs it is possible to calculate the steady-state size distribution of the product and capacity of a circuit that includes the crusher. These data are used to evaluate the fitness of proposed designs. Each evaluation takes approx 300 ms on a 700MHz Pentium III.

3 ALGORITHM

The problem described above is well suited to an evolutionary algorithm approach. The problem cannot easily be described analytically, but a simulation is available that can be used to evaluate candidate solutions. The search space is large - too large for an exhaustive search - and there is little to guide an engineer in determining good designs for a given scenario. We chose an evolution strategy (ES) approach to tackle this problem, as it has similarities with other problems that have previously been successfully handled by ES. In particular, candidate designs can be described using a vector of real values, and the problem involves determining geometric shapes. Previously reported successful applications of this type include the design of a jet nozzle (Klockgether, 1970) and a flywheel (Eby, 1999).

The basic evolution strategy algorithm has the following steps:

1. Create an initial population of designs.
2. Evaluate the fitness of these designs.
3. Create a population of children by mutating the members of the current population.
4. Evaluate the fitness of these children.

5. Select the fittest designs from the parents and children together.

6. Repeat steps 3 to 5 until done.

To implement a specific instantiation of the algorithm, we must specify the representation scheme to be used, the method of fitness evaluation, the nature of the mutation operators, the selection mechanism, and the termination condition. It may be possible for infeasible designs to be generated by mutation, in which case we must also specify how to deal with these infeasible designs.

These specifications are detailed in the remainder of this section.

3.1 FITNESS

The principal objective that we are trying to maximise is the capacity of a circuit containing a given crusher. The placement of the crusher in a circuit is important because a crusher that itself has a high capacity may not be suitable if it generates a lot of oversize material: the presence of this recirculating material reduces the rate at which feed can be introduced into the circuit. We define “capacity ratio” to be the ratio of the amount of material entering the crusher to the amount of feed entering the circuit (at steady-state operation). A higher capacity ratio corresponds to more recirculating material.

The capacity of a circuit may be limited by one of three factors.

1. The capacity of the crusher. If a crusher has capacity CAP tons/hour and capacity ratio CR , the capacity of the circuit will be limited by

$$CAP / CR$$

2. The power requirements of the crusher. A high rotational speed in particular delivers a lot of crushing but requires a lot of power. If a crusher with maximum power output MP kWh requires P kWh to process a circuit feed of F tons/hour, the capacity of the circuit will be limited by

$$F \times (MP / P)$$

3. The capacity of the recirculation conveyor in the circuit. If a crusher has capacity ratio CR and the conveyor has a capacity of MR tons/hour, the capacity of the circuit will be limited by

$$MR / (CR - 1)$$

Each of these factors potentially limits the capacity of the circuit, therefore the actual capacity will be the minimum of these values.

Notice the potential trade-offs for the various design variables. For example, a large closed-side setting will increase the capacity of the crusher, but will also increase the amount of recirculating material, raising the capacity ratio. Similarly, a high rotational speed will lead to more crushing in each pass through the chamber, but will also increase the power requirements of the crusher, possibly reducing the overall capacity.

Alongside maximising the capacity of the circuit, we also want to minimise the size of the product. Specifically, we define $P80$ to be a measure of the size of the 80th percentile in the product (i.e. the size k mm such that 80% of the product is smaller than k mm). For technical reasons, a higher value of $P80$ corresponds to a smaller product, so we want to maximise $P80$.

For the purpose of the experiments reported in this paper, we normalise both capacity and size figures by dividing by the figures for a standard design and settings.

In an earlier study, (Hingston, 2002), we combined the two objectives by defining the fitness of a design as a linear combination of them. The fitness function used in the earlier study was:

$$0.05 \times CAP + 0.95 \times P80$$

where CAP is the circuit capacity, $P80$ is the size measure, and the constants are chosen to equalise the variability of the two components. Thus the fitness of the standard design is 1.0, and higher fitness is better.

In the present study, we use both objectives to define the Pareto ranking of a design relative to a set of potential designs. We use the ranking scheme proposed by Fonseca and Fleming (Fonseca, 1998), as described in (Veldhuizen, 2000). We define Pareto dominance for designs as follows:

A design u is said to *dominate* a design v iff $CAP(u) > CAP(v)$ and $P80(u) > P80(v)$

A design x is *Pareto optimal* with respect to a set of designs Ω iff there is no design in Ω that dominates x . Thus a design that is Pareto optimal cannot be improved in any objective without degrading other objectives.

Finally, the *Pareto rank* of a design x , with respect to a set of designs Ω , is the number of designs in Ω which dominate x .

Thus x is Pareto optimal iff x has a Pareto rank of 0. In this multi-objective approach, Pareto rank, rather than a combined fitness value, is used as the basis for selection.

3.2 INITIALISATION

The population is initialised with copies of the existing standard design and settings. These copies are quickly eliminated in the first few generations of a typical execution.

3.3 REPRESENTATION

The representation of the machine settings - closed-side setting, eccentric angle and rotational speed - is straightforward, these being real values within given ranges. The best way to represent the geometric shapes of the two liners is less clear. The shape of each liner is defined by its vertical cross-section. The shape of the machine structure dictates the shape of the “back” of each liner, so it is only the “front” of each liner (the actual crushing surface) that is represented.

We chose to describe each shape as a series of line segments, using a variable-length list of points, each represented by a pair of coordinates. The first coordinate pair for the first segment and the last coordinate pair for the last segment are fixed, but each other coordinate is another real-valued object variable. Thus, if there are n line segments on the mantle and m line segments on the bowl liner, then the genotype consists of a vector of

$$3 + 2(n-1) + 2(m-1)$$

real-valued object variables.

Figure 3 shows a series of liner pairs evolved during a typical run.

3.4 MUTATION

When a parent is mutated to produce a child, each object variable is mutated independently using self-adaptive mutation rates as described in (Back, 1997). Specifically, each object variable is mutated using the formula

$$X'_i = X_i + \sigma'_i \cdot N_i(0,1)$$

where $N_i(0,1)$ is a normally distributed random value with mean 0 and standard deviation 1, and each strategy parameter σ_i is mutated using the formula

$$\sigma'_i = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1))$$

where τ and τ' are constants set to 0.25 and 0.1 respectively, and $N(0,1)$ is sampled once for each individual.

In addition, we provided mutation operators to increase or reduce the number of segments in a liner. Whether to apply these operators is determined randomly with a fixed probability. The mutation to reduce the number of segments randomly selects two adjacent segments to merge and discards the common end point. The operator to increase the number of segments randomly selects a segment to split into two, using the segment midpoint as the common end point. This was done to allow the algorithm to generate more complex or simpler liner shapes as required.

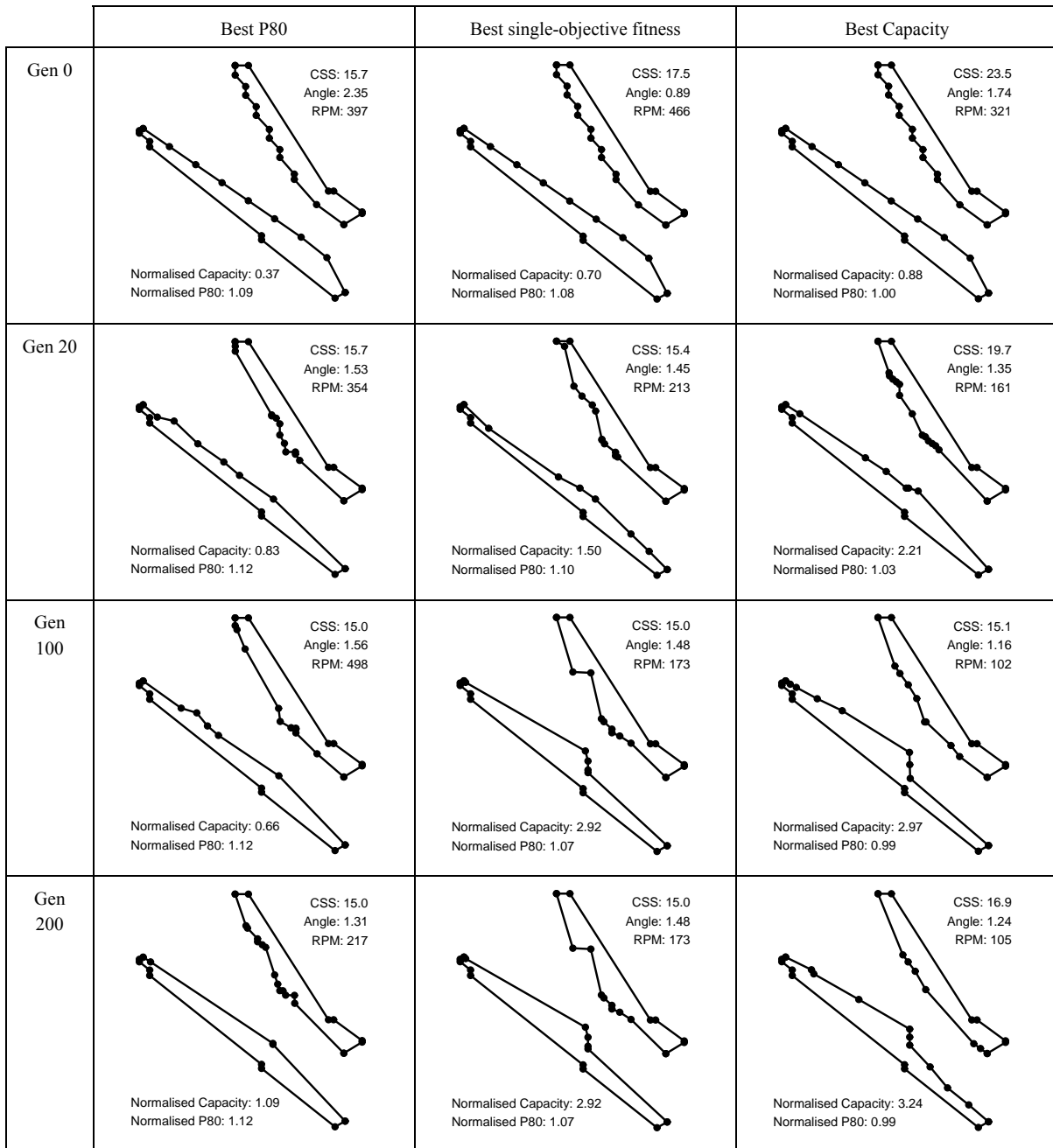


Figure 3 - A sample of evolved liner pairs

3.5 CONSTRAINTS

There are a variety of feasibility constraints upon potential designs. These can be categorised as follows:

Physical constraints The sequences of coordinate pairs must describe shapes that make sense operationally. In particular, the liners must have at least a certain thickness to be practical. We found that this constraint was violated so rarely that it is not worth the computational expense to do the checking. If the final solution returned violates this constraint, the algorithm can simply be re-run.

Setting constraints Each machine setting must be confined to a given range. This is done by repair — any value that is too low is set to the minimum value for that setting, and any that is too high is set to the maximum value.

Modeling constraints The crusher simulation is very complex and assumes (sometimes implicitly) that liners have “sensible” shapes. To keep our designs in the “sensible” region, we imposed a heuristic constraint that the sequence of x-coordinates and the sequence of y-coordinates for each liner always change monotonically.

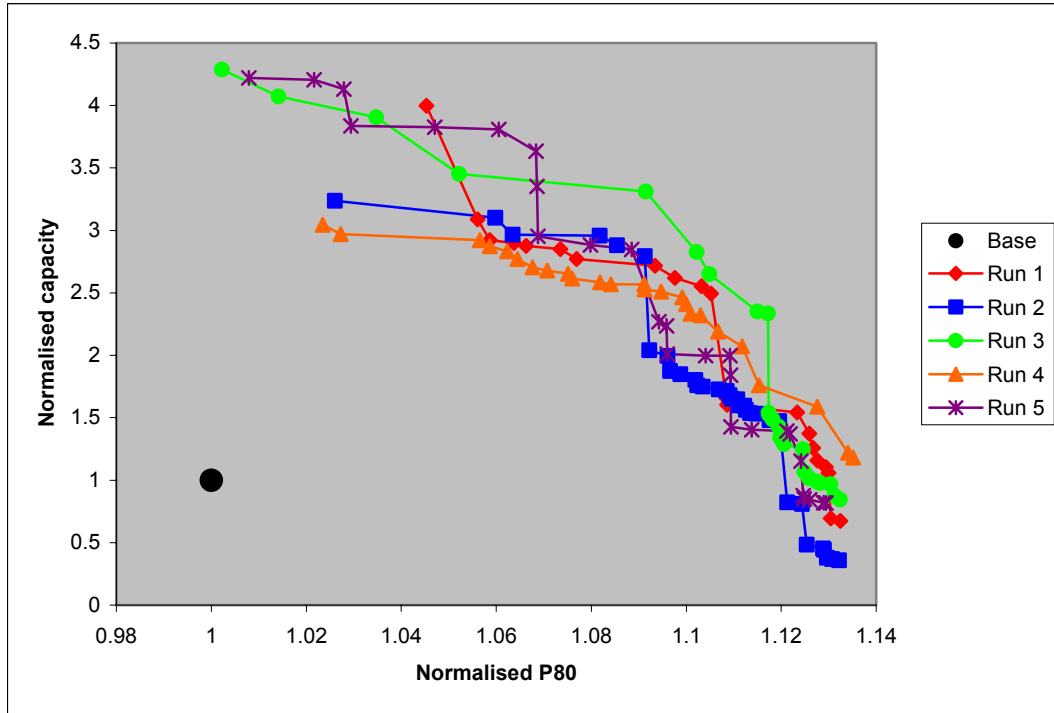


Figure 4 - Final Pareto fronts from five runs of the system

This constraint is enforced by repairing any coordinate that violates the constraint, at the time of creation. Even so, the simulation occasionally fails. In these cases, the design is assumed to be nonsensical and both capacity and $P80$ are assigned an abysmal value of 0.

3.6 SELECTION

Selection is done using the standard $(\lambda + \mu)$ -selection mechanism of evolution strategies, with $\lambda = \mu = 1$. Each member of the current generation becomes the parent of one child, and those with lowest rank are selected from the parents and children combined become the next generation. That is, each member of the current generation becomes the parent of one child, and the best individuals selected from the combined parents and children become the next generation.

4 RESULTS AND DISCUSSION

In this section, we describe an example set of runs of the system that is indicative of the performance on test problems. We ran the system five times with a population size of 50 for 200 generations on each run.

Figure 4 shows the final Pareto fronts for the sample runs. In all cases a good range of designs is found, showing different tradeoffs of the two objectives.

Figure 5 shows the movement of the Pareto front during Run 5 from Figure 4. Note that while the fronts for Generations 100 and 200 appear to cross, no design in Generation 200 is actually dominated by one in

Generation 100. Some designs in Generation 100 are still present in Generation 200 (indeed one design in Generation 20 is still present), and the use of lines to interpolate between the population members creates the illusion of a cross-over. The situation is exacerbated by the difficulty in improving $P80$ values beyond a certain level: this has been confirmed by experiments where maximising $P80$ was the sole objective.

Figure 3 shows a sample of evolved liner pairs and settings from another run. The first row shows liners from Generation 0, a selection of random mutations on the standard design. The middle rows show liners from part way through the run, and the final row shows liners from the final Pareto front. The first column shows the design with the best $P80$, while the last column shows the design with the best capacity. The second column shows the design with the highest fitness according to the composite measure used in (Hingston, 2002).

Figure 6 shows the user interface during the execution of a typical run. The top right corner shows a scatter plot of the current generation in objective-space. The user can select a particular generation design in the plot to view its details elsewhere on the screen. The top left corner depicts the selected crusher in a circuit: it shows the liner shape and the material flows through each part of the circuit. The user can click on one of these flows to view a graph of the size distribution of the corresponding ore stream. The bottom left corner shows the settings and fitness for the selected design. The bottom right corner has various controls for the parameters of the system.

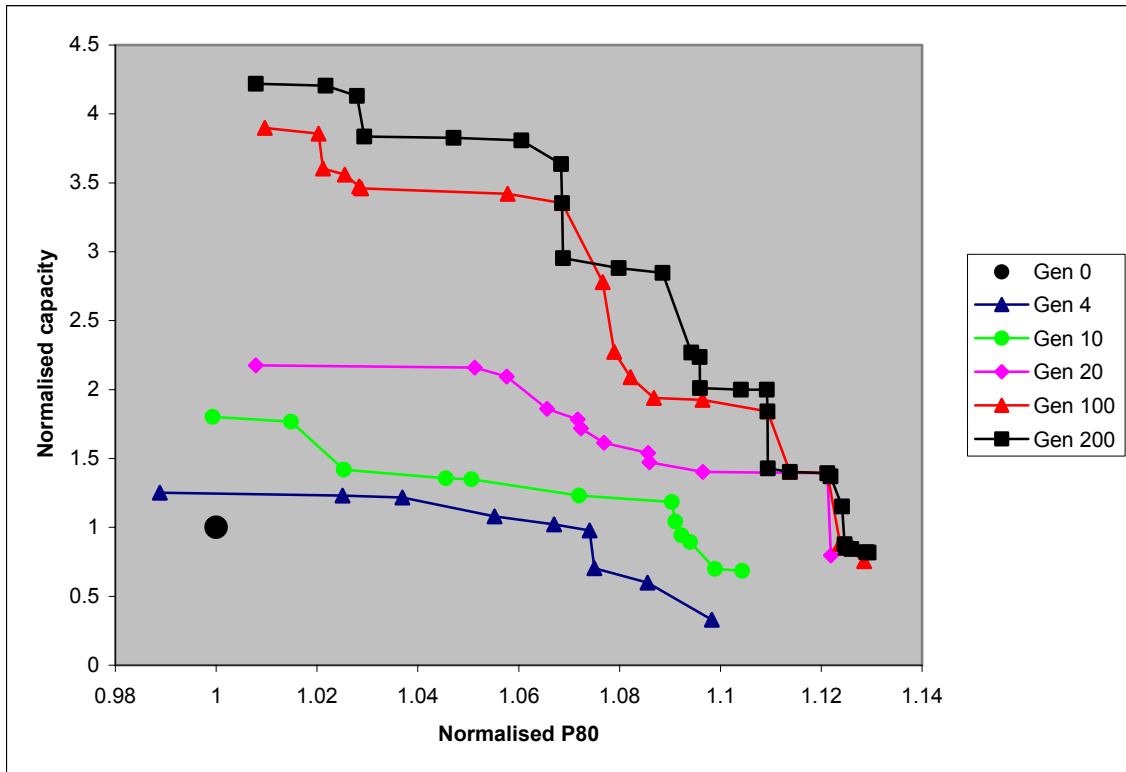


Figure 5 - Progressive Pareto fronts from Run 5 in Figure 4

The true multi-objective approach used in this study offers clear benefits in this application over the simpler approach of using a combined fitness function as in (Hingston, 2002). It removes the need for arbitrary weightings, which engineers have trouble specifying in advance. There is no need to separately apply capacity constraints, as non-dominated solutions inevitably satisfy the constraints anyway. The user interface provides an intuitive visualisation for engineers, enabling them to see the effects of trading off the different objectives on the evolved designs.

5 FUTURE WORK

The work reported here is still in the early stages of its development. While the results obtained so far are excellent, many enhancements and extensions are envisaged.

Planned enhancements to the crusher simulation are likely to make it run an order of magnitude slower. We may then need to develop special strategies to speed up the evolutionary algorithm. One possibility is to use faster, more approximate models early in the search, using a scheme similar to the injection island genetic algorithm described in (Eby, 1999).

Another aim is to include, as part of the task, the design of the circuit itself - that is, to co-evolve crushers, screens and other processing units and their settings, as well as the pattern of conveyors connecting them together. This

brings in elements of network design, another application area in which evolutionary algorithms have been successful (see e.g. (Gross, 1996)). The concurrent design of this network and the machines within it will be challenging, but the potential rewards are huge.

6 CONCLUSIONS

In this paper we have described a study in the application of multi-objective evolutionary algorithms to a difficult practical engineering design problem. Our system determines the liner profiles and operating settings for a crusher in a comminution circuit. Initial results promise significant financial benefits.

In many ways, this problem is an ideal application for evolutionary algorithms - the pay-off is high; the problem is too complex to solve analytically; the search space is too large to explore unaided; we have a well defined evaluation function and a straightforward representation scheme, suitable for manipulation by genetic operators. Many challenges remain in incorporating more realism in the problem definition (for example, including variety in feed properties, interactions with other plant etc) and validating the predicted performance with field trials.

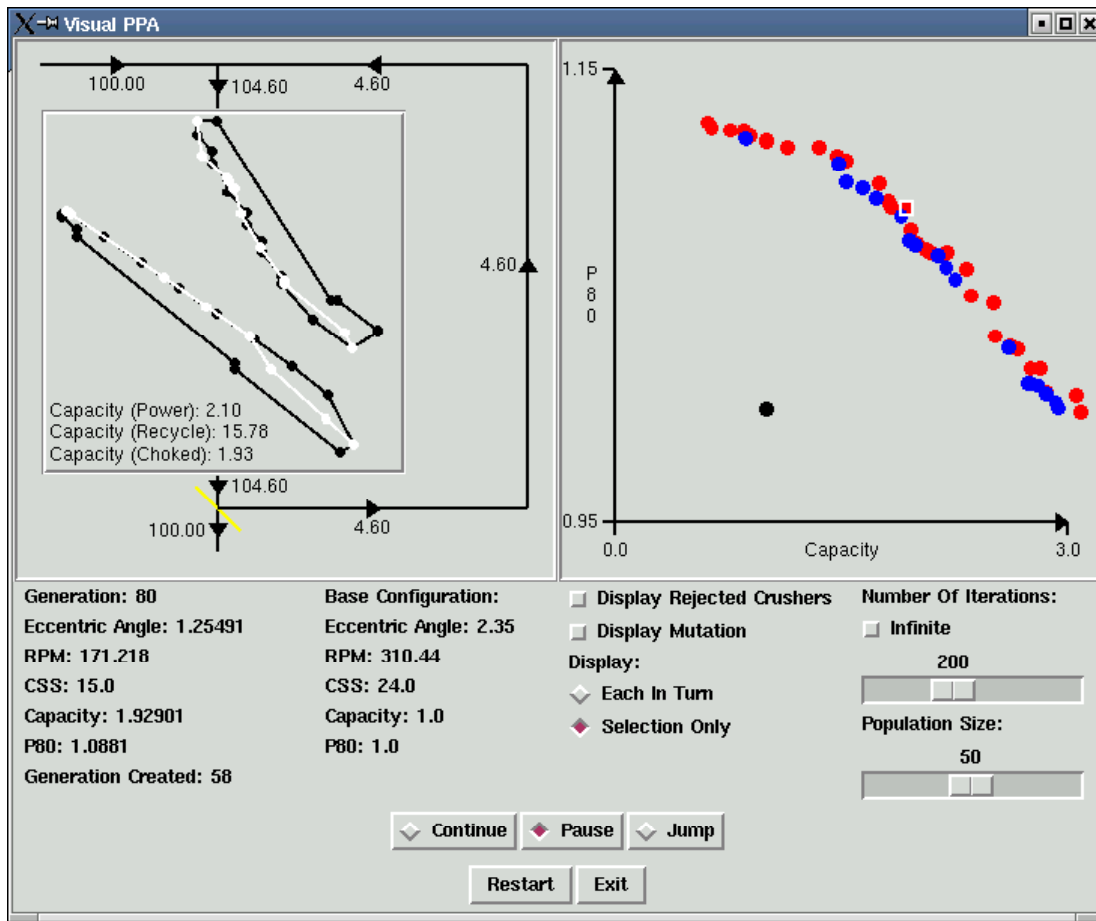


Figure 6 - The user interface of the system

References

- Back, T., Ulrich, H. & Schwefel, H-P. (1997). Evolutionary Computation: Comments on the History and Current State. *IEEE Transactions on Evolutionary Computation*, 1(1), 3-16.
- Eby, D., Averill, R.C., Punch, W.F. & Goodman, E.D. (1999). Optimal Design of Flywheels Using an Injection Island GA. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing*, 13, 327-340.
- Fonseca, C. M., & Fleming, P.J. (1998). Multiobjective Optimisation and Multiple Constraint Handling with Evolutionary Algorithms - Part I: Unified Formulation. *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, 28(1), 26-37.
- Gross, B., Hammel, U., Maldaner, P., Meyer, A., Roosen, P. & Schutz, M. (1996). *Optimization of Heat Exchanger Networks by Means of Evolution Strategies*. Paper presented at the Sixth Conference on Parallel Problem Solving from Nature.
- Hingston, P., Barone, L. & While, L. (2002). *Evolving Crushers*. Paper presented at the Conference on Evolutionary Computation, Hawaii, 2002.
- Hiorns, F. J. (1971). Wear measurements in size reduction machinery. *Chemical Process Engineering*(January), 47-49.
- Klockgether, J. S., H-P. (1970). *Two-phase Nozzle and Hollow Core Jet Experiments*. Paper presented at the 11th Symposium on Engineering Aspects of Magnetohydrodynamics.
- Napier-Munn, T. J., Morrell, S., Morrison, R.D. & Kojovic, T. (1996). *Mineral Comminution Circuits*. Brisbane: Julius Kruttschnitt Mineral Research Centre.
- Veldhuizen, D., & Lamont, G. (2000). Multiobjective Evolutionary Algorithms: Analysing the State-of-the-Art. *Evolutionary Computation*, 8(2), 125-147.