2002

Evolving Crushers

Philip Hingston
Edith Cowan University

Luigi Barone
University of Western Australia

Ronald While
University of Western Australia

10.1109/CEC.2002.1004398
This is an Author's Accepted Manuscript of: Hingston, P. F., Barone, L. & While, R. (2002). Evolving Crushers. Proceedings of Congress on Evolutionary Computation. (pp. 1109 - 1114). Honolulu, USA. IEEE. Available here

Evolving Crushers

P. Hingston
School of Computer and Information Science
Edith Cowan University
Mt Lawley, WA, Australia

L. Barone **L. While**
Department of Computer Science & Software Engineering
The University of Western Australia
Nedlands, WA, Australia

Abstract — This paper describes the use of an evolutionary algorithm to solve an engineering design problem. The problem involves determining the geometry and operating settings for a crusher in a comminution circuit for ore processing. The intention is to provide a tool for consulting engineers that can be used to explore candidate designs for various scenarios. The algorithm has proved capable of deriving designs that are clearly superior to existing designs, promising significant financial benefits.

***Keywords*: Evolutionary algorithms, evolution strategies, engineering design.**

I. 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 ore processing. This study clearly demonstrates the strengths of the evolutionary approach.

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.

We begin the paper with a description of the problem, including a brief background on crushers and comminution circuits. Section III describes our mapping of the problem to an evolutionary algorithm, including the genetic representation, genetic operators and selection methods. Section IV 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.

II. 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 ([1]). Thus optimisation of crushing operations offers large potential economic benefits. For example, in the area of energy savings, Napier-Munn et al ([2], 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.

A. 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, [2] 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.

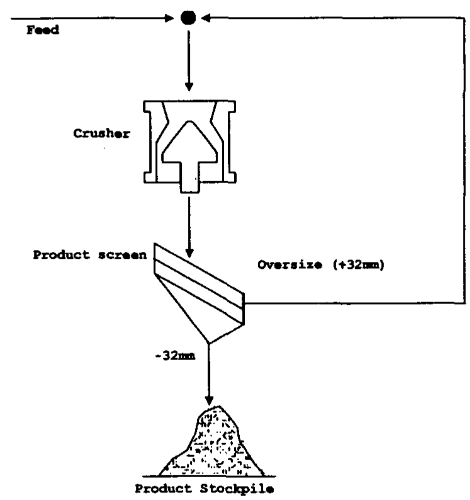

Figure 1 - The simple circuit used in this study

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.

B. *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.

The final four of these were chosen 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 chamber without overloading the crusher (its "capacity").

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

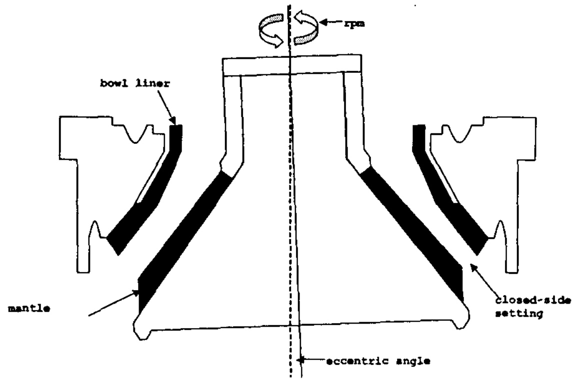

Figure 2 - Schematic diagram of a cone crusher (after [2] Figure 6.3)

III. 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 approach to tackle this problem, as it has similarities with other problems that have been successfully handled by evolution strategies. 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 ([3]) and a flywheel ([4]).

The basic evolution strategy algorithm has the following steps:

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

A. 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.

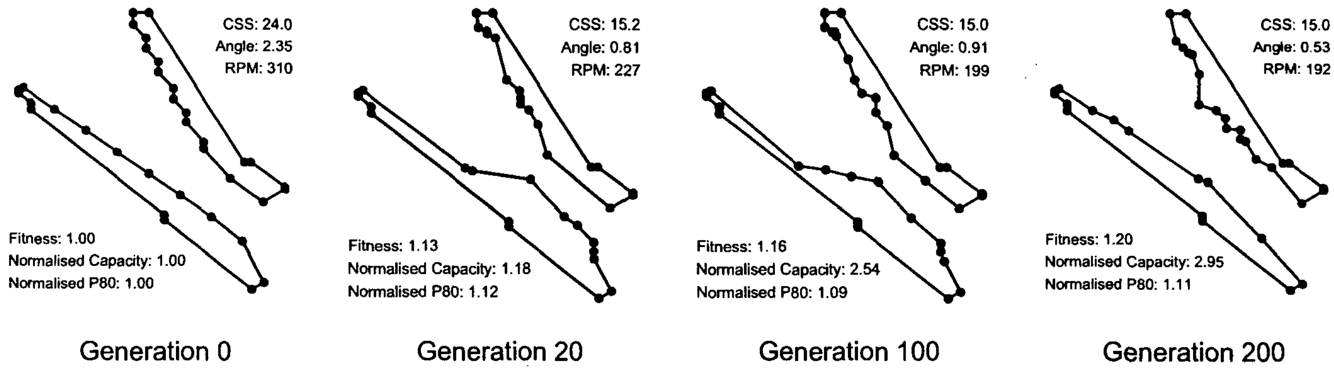

Figure 3 - A series of evolved liner pairs

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.

A secondary aim of the process is 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.

The actual fitness function that we use is:

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

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.

B. 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.

C. 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. The first pair is a standard design as might be supplied by a crusher manufacturer.

D. Mutation

When a parent is mutated to produce a child, each object variable is mutated independently using self-adaptive mutation rates as described in [5]. 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

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

where τ and τ' are constants set to 0.25 and 0.1 respectively. $N(0,1)$ 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 operator 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 desired.

E. 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. Whilst code was developed to enforce this constraint, we found that it is 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 (sometime 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 must both change monotonically. 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 is assigned an abysmal fitness of 0.

F. Selection

Selection is done using the standard ($\lambda + \mu$)-selection mechanism of evolution strategies, with $\lambda = \mu = 1$. 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.

IV. RESULTS AND DISCUSSION

In this section, we describe an example set of runs of the algorithm that is indicative of the performance attained on test problems.

We ran the system ten times with a population size of 50 for 200 generations on each run. Table 1 shows the performances of the best designs from these runs. The results show an average increase in capacity of around 140%, and around 10% in *P80*.

TABLE 1 - PERFORMANCES OF THE BEST DESIGNS FROM TEN RUNS RELATIVE TO THE STANDARD DESIGN.

Run	Capacity	P80	Fitness
1	2.068	1.106	1.154
2	2.176	1.106	1.160
3	1.981	1.108	1.152
4	2.288	1.101	1.160
5	3.087	1.094	1.194
6	1.781	1.117	1.150
7	1.958	1.106	1.149
8	3.065	1.093	1.191
9	2.591	1.105	1.180
10	2.947	1.107	1.199

Figure 4 shows how the fitness values and the two components, *P80* and capacity, evolve during a typical run, Run 10. Improvements in capacity have been scaled down by a factor of 19 to reflect the fitness function scaling. It can be seen that improvements tend to be made by favourable tradeoffs between the two components.

Figure 3 shows the best liner pairs from selected generations evolved during another run. It can be seen that the evolved shapes are distinctly different from the standard design. Whilst engineers can provide a post-hoc rationale for the revised design, and this provides confidence in the validity of the designs, it is virtually impossible to predict in advance the effect of a change in shape, much less to intuit a high quality design for a specific scenario.

It is worth noting that each run takes only around 30 minutes. In a real design exercise, a running time of several hours (or even days) would still be very acceptable, so there is plenty of scope for increased task complexity in the future.

V. 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. The problem described in this study could be extended to include other objectives. Work has begun on a multi-objective algorithm based on Pareto optimality, using the principles outlined in [6].

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 [4].

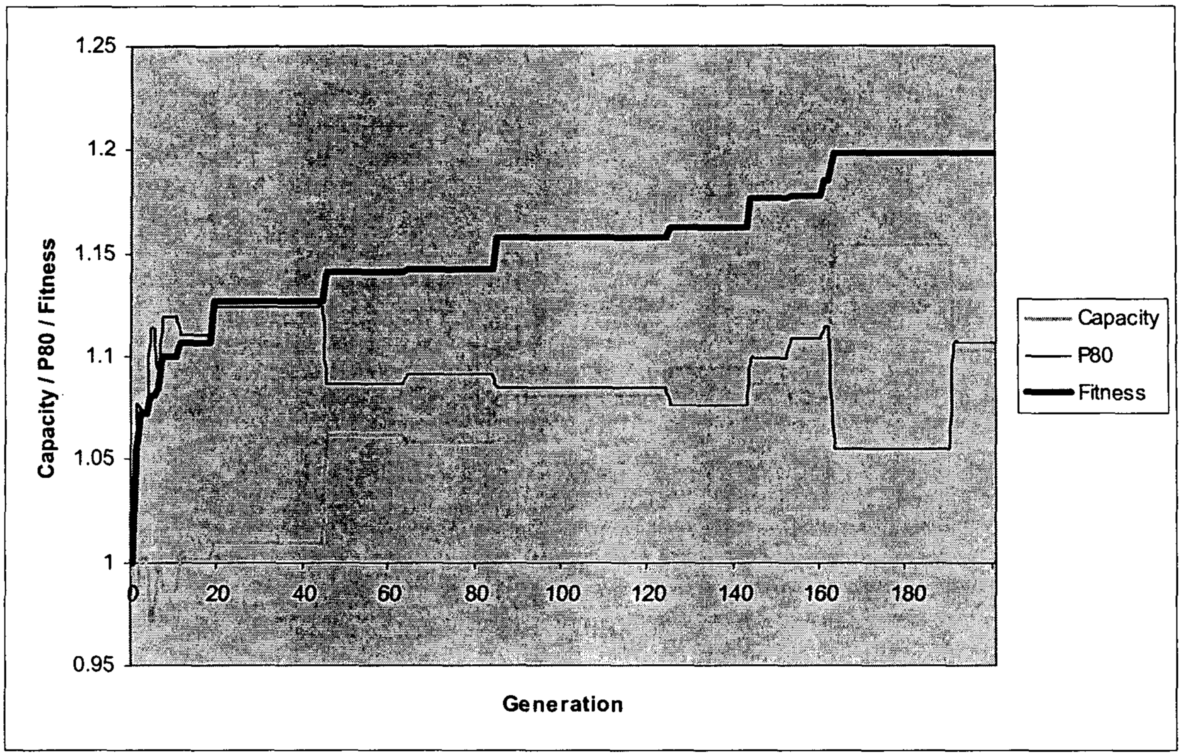

Figure 4 - Graph showing fitness evolution during Run 10 from Table 1

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. [7]). The concurrent design of this network and the machines within it will be challenging, but the potential rewards are huge.

VI. CONCLUSION

In this paper we have described a study in the application of evolutionary algorithms to a difficult practical engineering design problem. Our system determines the liner profiles and operating settings for a comminution circuit in an ore processing plant. Initial results are very promising and indicate 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.

VII. ACKNOWLEDGEMENTS

This work was supported by Rio Tinto Research Grant BTC005. The authors would like to acknowledge the support and assistance of Rio Tinto Technical Services.

VIII. REFERENCES

[1] F. J. Hiorns, "Wear measurements in size reduction machinery," *Chemical Process Engineering*, pp. 47-49, 1971.

[2] T. J. Napier-Munn, Morrell, S., Morrison, R.D. & Kojovic, T., *Mineral Comminution Circuits*. Brisbane: Julius Kruttschnitt Mineral Research Centre, 1996.

[3] J. S. Klockgether, H-P., "Two-phase Nozzle and Hollow Core Jet Experiments," presented at 11th Symposium on Engineering Aspects of Magnetohydrodynamics, 1970.

[4] D. Eby, Averill, R.C., Punch, W.F. & Goodman, E.D., "Optimal Design of Flywheels Using an Injection Island GA," *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing*, vol. 13, pp. 327-340, 1999.

[5] T. Back, Ulrich, H. & Schwefel, H-P., "Evolutionary Computation: Comments on the History and Current State," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 3-16, 1997.

[6] D. Veldhuizen, & Lamont, G, "Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art," *Evolutionary Computation*, vol. 8, pp. 125-147, 2000.

[7] B. Gross, Hammel, U., Maldaner, P., Meyer, A., Roosen, P. & Schutz, M., "Optimization of Heat Exchanger Networks by Means of Evolution Strategies," presented at The Sixth Conference on Parallel Problem Solving from Nature, 1996.