Journal of Heuristics, 10: 153–167, 2004

An Iterative Construction Heuristic for the Ore Selection Problem

A.J. RICHMOND
Earth Science and Engineering Department, Imperial College, London, SW7 2AZ, UK
email: a.richmond@imperial.ac.uk

J.E. BEASLEY
The Management School, Imperial College, London, SW7 2AZ, UK
email: j.beasley@imperial.ac.uk

Submitted in February 2003 and accepted by Anthony Cox, Jr. in February 2004 after 1 revision

Abstract

The ore selection problem involves choosing a processing option for a number of mining blocks that maximises the expected payoff for a given level of financial risk. An innovative neighbourhood search heuristic is proposed for the ore selection problem. This iterative construction heuristic employs a stochastic demolition and reconstruction strategy. Computational experiments with this heuristic for two ore selection problem instances, one involving 2,500 blocks and the other involving 78,000 blocks, are given. These problem instances are made publicly available for use by future workers. Our computational experiments indicate that the proposed heuristic produces better quality solutions faster than a relay hybrid (constructive-simulated annealing) heuristic.

Key Words: combinatorial optimisation, portfolio theory, downside risk, ore selection problem

1. The ore selection problem

In open pit (cast) mining, rock material is extracted from the Earth's surface and processed in order to produce mineral commodities profitably. Prior to mining, the area to be mined can be conveniently thought of as being divided into discrete, non-overlapping, three-dimensional rectangular blocks. The usual assumption is that each block must be mined, i.e. that the material (typically rock) in each block must be physically dug out of the Earth. In terms of processing the material from each block, a decision must be made as to whether to discard that material as waste, or to process it as ore to extract value (mineral commodities). The reason why a block is mined, even if its material is discarded as waste, is that the material must be physically moved in order to gain access to other blocks which are more valuable.

A number of ore processing options, with varying degrees of mineral recovery and processing cost, may exist. Thus, even if a block is selected as ore, a decision as to the mode of processing must be made. The problem of whether to treat a particular block as waste or ore (and the processing option to be used) is not a simple one as the true mineral content of the block is not known. Instead, sample information (blast holes assays) is typically used

to provide estimates of the mineral content of any block. These estimates in turn provide financial estimates of the profit that could be obtained from each block.

Hence in this paper we are considering a discrete decision problem under uncertainty:

- discrete—as for each block we need to decide whether to treat it as waste or as ore (and if as ore which processing option should be adopted)
- uncertainty—as we do not know the exact financial value of each block.

Traditional methods adopted in the mining industry for dealing with this problem focus on approaches such as cut-off grades (Lane, 1988), loss functions (Srivastava, 1987), and utility functions (Richmond, 2002), where each block is considered independently. In such approaches the decision as to how to treat a particular block can be made without reference to the decisions made for other blocks. The key disadvantage of these methods is that they do not take into account the potential diversification benefits from considering the set of decisions as a whole (Richmond, 2003). The diversification of financial risk by suitable choice of block processing options can be formulated (as below) as a constrained optimisation problem that we will refer to as the ore selection problem (OSP).

Let:

K be the number of mutually exclusive and exhaustive processing options, where without loss of generality we include treating a block as waste as one of the processing options
p_k be the proportion ($0 \leq p_k \leq 1$) of mineral recovered using processing option $k, k = 1, \ldots, K$
c_k be the mining and processing cost per unit weight for processing option $k, k = 1, \ldots, K$
B be the total number of blocks
q_b be the quantity (weight) of material contained in block $b, b = 1, \ldots, B$
P_b represent the set of possible processing options for block b, so that $P_b \subseteq [1, \ldots, K]$, $b = 1, \ldots, B$.

The local financial payoff (profit) $w(b \mid k)$ associated with option $k \in P_b$ for block b ($b = 1, \ldots, B$) is calculated as:

$$w(b \mid k) = vp_k[z(b)q_b] - c_k q_b \quad \forall k \in P_b, \quad b = 1, \ldots, B \tag{1}$$

where v is the mineral value per concentration unit and $z(b)$ is the mineral concentration per unit weight for block $b(b = 1, \ldots, B)$. In Eq. (1) $z(b)q_b$ represents the total quantity of the mineral in block $b(b = 1, \ldots, B)$, of which a proportion $p_k(k \in P_b)$ is recovered.

The true values of all parameters in Eq. (1) are unknown and contribute to financial uncertainty. However, the dominant source of financial uncertainty is related to uncertainty in the $z(b)$ values. In practice, q_b is determined from rock density estimates and is assumed to be constant for all blocks of similar rock type and dimension, and p_k and c_k are identified from previously processed blocks from the same (or a similar) mine. Mineral values v can be assumed constant at current values or uncertainty minimised through the use of forward selling contracts. Note that in Eq. (1) the financial payoff for a block is conditional on the processing option k chosen.

Determination of the (spatially dependent) $z(b)$ values relies on geostatistical approaches that are complex and far beyond the scope of this paper (e.g. Goovaerts, 1997, pp. 369–436). We will simply say here that a multivariate density function that characterises the joint uncertainty with respect to the B values $z(1), \ldots, z(B)$ is derived via conditional simulation, an approach to deriving a density function that accords with sampled data. In practice, between ten and one hundred conditional simulations are typically used to define the space of uncertainty in mineral concentration values. Combining the multivariate density function of mineral concentrations with Eq. (1) it is possible to derive the cumulative distribution of total financial payoff given the block processing options chosen for the entire set of blocks.

Let:

$y_{bk} = 1$ if we decide to process block b ($b = 1, \ldots, B$) using option $k \in P_b = 0$ otherwise
S be the block processing options chosen for the entire set of blocks so that $\mathrm{S} = [(b, k) \mid (y_{bk} = 1, k \in P_b), b = 1, \ldots, B]$
F_S be the cumulative distribution of total financial payoff given the set S of block processing options chosen for the entire set of blocks
R^* be the desired financial risk

then the OSP is:

$$\text{maximise} \quad E\{F_S\} \tag{2}$$

subject to

$$r\{F_S\} = R^* \tag{3}$$

$$\sum_{k \in P_b} y_{bk} = 1 \quad b = 1, \ldots, B \tag{4}$$

$$S = [(b, k) \mid (y_{bk} = 1, k \in P_b), b = 1, \ldots, B] \tag{5}$$

$$y_{bk} \in (0, 1) \quad \forall k \in P_b, b = 1, \ldots, B \tag{6}$$

where $E\{\cdot\}$ is the usual expectation operator and $r\{\cdot\}$ is a real valued risk function. Equation (2) maximises the expected payoff of the set of decisions. Equation (3) ensures that the set of decisions have the desired financial risk R^*. Equation (4) ensures that exactly one processing option is chosen for each block and Eq. (5) defines the set of decisions S. Equation (6) indicates that the problem is discrete.

In a modern portfolio theory context, solutions to the OSP for various R^* identify the constrained efficient frontier. In other words a discrete efficient frontier can be constructed by solving OSP (Eqs. (2)–(6)) for varying R^* values, eliminating any solutions that are dominated, and thereby generating a discrete curve showing the best possible tradeoff between expected payoff and financial risk. A similar constrained portfolio optimisation problem related to stock (equity) markets is discussed by Chang et al. (2000).

There are several different approaches to measuring risk, such as variance, semi-variance, expected loss, and probability of loss. In mining, measuring financial risk by the probability

weighted dispersion of potential payoffs below a target is appealing since it recognises the finite non-renewable nature of the resource in question and the desire to achieve a minimum acceptable payoff from the resource (Richmond, 2003). In this paper, the downside risk function proposed by Fishburn (1977) is adopted as a suitable measure of risk. This downside risk function is:

$$r\{F_S\} = \int_{-\infty}^{t} (t - w)^{\alpha} dF_S \tag{7}$$

where t is the target payoff and α (≥ 0) is a measure of the impact of failing to reach the target payoff t.

As formulated above the OSP (Eqs. (2)–(7)) is a zero-one non-linear problem that is hard to solve optimally. In order to approach this problem heuristically we will amend the objective function (Eq. (2)) to be a weighted sum of expected payoff and risk. Let λ ($0 \leq \lambda \leq 1$) be the weight to be attached to expected payoff, then the (amended) OSP that we consider in this paper is:

$$\text{maximise} \quad \lambda E\{F_S\} - (1 - \lambda) r\{F_S\} \tag{8}$$

subject to Eqs. (4)–(7).

2. Constructive heuristics

A constructive (successive augmentation) heuristic (CH) starts from a seed component. Then, other components are selected and added to (inserted into) the partial solution until a complete solution is obtained. It is common in the literature that each component is added/inserted by considering its influence on a solution's objective function. The purpose of CH's is to find a feasible solution by some simple best-effort strategy without excessive computation time (Altenbernd and Hansson, 1998).

Applied independently, CH's are very fast but suffer from the risk of finding solutions that are not locally optimal. Thus, most practical applications of CH's involve finding feasible solutions that are then used as initial solutions for iterative neighbourhood search heuristics such as simulated annealing or tabu search. To overcome the failure of CH's in finding local optima without resorting to these high-level relay hybrid heuristics (Talbi, 2002), we propose a greedy iterative construction heuristic (ICH) that employs a stochastic demolition and reconstruction strategy. ICH is only appropriate for solving classes of problems in which the random extraction and insertion of components into partial solutions can be guaranteed to lead to a feasible solution. An instance of this class is the OSP described above. Section 3 presents our iterative constructive heuristic in the context of the OSP. We illustrate its working in Section 4 using a practical example. In Section 5 we report results of computational experiments for OSP's of various size. To get an indication of the promise of the proposed heuristic, we compare the use of ICH against a constructive-simulated annealing heuristic. We summarise the paper in Section 6.

Let S be an ordered set of B elements where the b'th element of S takes the value zero if no processing option has been chosen for block b, otherwise it takes some value $k \in P_b$ indicating that option k has been chosen for block b (b=1,...,B). We denote the objective function (equation (8)) value associated with S by $V(S)$ with the proviso that any blocks b for which no processing option has been chosen in S are excluded from that calculation.

```
1    Randomly generate a feasible initial solution S
2    Select a stopping criterion parameter value I
3    Select the initial value of J where 1 ≤ J ≤ B
4    Set the stopping criterion counter ∂=-I
5    Set the iteration counter C=0
6    Repeat
7        Set C=C+1
8        If C=1 then
9            m=B
10       Else
11           Randomly generate an integer m∈[1,J]
12       Endif
13       Randomly select m distinct elements (blocks) in S and set their associated
             processing options to zero to get S#
14       Calculate the objective function for the partial solution S#
15       Repeat
16           Randomly select a block b from S# for which no processing option
                 has been decided
17           Examine all possible processing options for block b and choose the
                 option k∈Pb which provides the most beneficial change in
                 the objective function associated with S#
18           Add this decision to S# (set the b'th element of S# to k)
19           Update the objective function for the partial solution S#
20       Until a processing option has been decided for all blocks in S#
21       If V(S#) > V(S) then
22           S=S#
23           ∂=∂-I
24       Else
25           ∂=∂+1
26       Endif
27       If required, reduce the value of J
28   Until ∂≥I whereupon S, of value V(S), is the best solution found in C iterations
```

Figure 1. Pseudocode for our iterative constructive heuristic.

3. Description of the heuristic

The pseudocode for ICH is shown in figure 1. As an overview, ICH:

(a) starts from a randomly generated initial solution (line 1 in figure 1)
(b) randomly chooses the number of blocks (m) that are to have their processing decision re-examined (lines 8–12 in figure 1)
(c) stochastically selects m blocks and eliminates their processing decisions from the current solution (line 13 in figure 1)

(d) repeatedly (lines 15–20 in figure 1):

- stochastically selects a block b with no processing decision (line 16 in figure 1)
- chooses a processing option for block b using the deterministic rule that the processing option chosen best improves the objective function value (line 17 in figure 1)

(e) until (line 20 in figure 1) all blocks have a processing decision, whereupon (lines 21–26 in figure 1) we accept this new solution if it is better than the previous solution
(f) the above, steps (b) to (e), being repeated until (line 28 in figure 1) some user-defined stopping criteria is satisfied

We refer to this strategy as demolition (line 13 in figure 1) and reconstruction (lines 15–20 in figure 1). Reconstructed solutions are accepted only if the objective function is improved (lines 21–22 in figure 1), thus, ICH is a greedy heuristic.

The stopping criterion (line 28 in figure 1) used in our heuristic has been adopted from the stochastic evolution algorithm (Saab and Rao, 1991) to reward the heuristic with additional iterations whenever improved solutions are found (line 23 in figure 1). The parameter I represents the expected number of iterations the heuristic needs to find an improvement in the objective function with respect to the current solution. If I is too small, then the algorithm may not have sufficient time to improve the solution, and if too large the heuristic may waste too much time in the later stages of the search (Saab and Rao, 1991).

Computationally we observed that as ICH progresses it tends to make the same decision when considering an individual block b, irrespective of the sequence used in considering the m blocks. Thus, during later iterations, a reconstruction sequence that results in a different solution may not occur. The fact that repeated demolition and reconstruction does not alter the current solution implies only that the algorithm has converged (possibly to a local optimum), not that the current solution is locally optimal.

Referring to figure 1, we have that the initial solution is always completely demolished during the first iteration ($C = 1$) as m is set equal to B (line 9 in figure 1). If $I = 0$ the algorithm is simply a CH as it stops at the end of the first iteration. For values of $I > 0$, the algorithm becomes iterative, repeatedly demolishing and reconstructing m random block processing decisions in the current solution.

Regarding the pseudocode in figure 1 we have that (except for the first iteration) the number of blocks for which processing decisions need to be re-examined (demolished and reconstructed) is m, which is randomly generated from $[1, J]$ (line 11 in figure 1). However m need not be generated in such a way that all outcomes are equally likely. Letting s_m be the probability of choosing m blocks we have that a number of strategies can be followed:

Strategy (1) $J = B$ and all outcomes are equally likely, i.e. $s_m = 1/J, m = 1, \ldots, J$
Strategy (2) $J = B$ and s_m is inversely proportional to m, i.e. $s_m = (1/m)/(\sum_{b=1}^{J}(1/b))$, $m = 1, \ldots, J$
Strategy (3) Strategy (1) or (2) but with $J < B$
Strategy (4) Strategy (1) or (2) but with J reducing in a systematic manner from its initial value of B depending on structural changes in the solution.

Table 1. Effect of different m generation strategies on ICH execution.

Implementation	Strategy	J[a]	# iterations C	Time (s)	Quality
1	1	2,500	1,744	237	0.3662
2	2	2,500	3,487	114	0.0016
3	1 + 3	2	10,313	2	0.0071
4	1 + 3	10	8,548	5	0.0006
5	1 + 3	100	4,516	25	0.0021
6	2 + 3	2	10,606	2	0.0925
7	2 + 3	10	9,367	4	0.0000
8	2 + 3	100	5,104	11	0.0007
9	1 + 4	2,500 → 92	3,088	33	0.0006
10	2 + 4	2,500 → 51	3,487	7	0.0000

[a]the → indicates that J decreases during this implementation.

The first strategy equates to m being drawn randomly from a uniform distribution $[1, B]$. Thus, on average, $(B + 1)/2$ blocks are demolished and reconstructed at each iteration. For later iterations, we observed that this often involves considerable computational effort with no difference between the current and reconstructed solutions. The practical implementation of ICH for the OSP can be considerably enhanced computationally by considering the alternative strategies for generating m outlined above. This is demonstrated by the results of ten implementations of ICH, shown in Table 1, which illustrate the effect of using different strategies for generating m. The results in that table are for a test problem involving 2,500 blocks that we will consider in more detail in the following section. An "ideal" heuristic should produce a good quality solution in a short period of time. In this paper, solution quality is measured as:

$$100[(V(S^+) - V(S))/V(S^+)] \tag{9}$$

where S^+ is the best solution found over all ten implementations of ICH shown in Table 1. Equation (9), therefore, measures percentage deviation from the value of that solution, $V(S^+)$. The lower the quality value the better the corresponding solution. A quality value of zero indicates that the best known solution was found. The computation times shown in Table 1 are for our heuristic as coded in FORTRAN and run on a PC with a 1.7 GHz processor.

With regard to Table 1, we can see that if m is allowed to be drawn from $[1, B]$ for all iterations (Implementations 1 and 2 in Table 1), then the number of iterations is comparatively small. However, the solution quality tends to be poor and the time to completion excessive. If m is overly restricted (Implementations 3 and 6), then the time to completion is rapid, but the solution quality is poor. For this example, good solutions can be generated in a reasonable amount of time either by using values of J equal to 10 or 100 (Implementations 4, 5, 7 and 8), or by reducing J as the number of iterations C increases (Implementations 9 and 10).

Rerun and backtracking strategies that commonly appear in the literature are special cases of the proposed demolition and reconstruction strategy. A rerun strategy (also referred to as a replication or restart or multi-start strategy), where the best solution is found via multiple constructions, is equivalent to demolition of the entire solution at each iteration. Backtracking to a prior partial solution to try a different strategy is equivalent to demolishing the local decisions in reverse order at each iteration. Introducing a stochastic element into backtracking has been proposed by Langley (1992) and Prestwich (2000), however, their goal was simply to generate feasible solutions.

The demolition and reconstruction strategy used in our ICH algorithm can be related both to repair algorithms and to macro-mutations in genetic algorithms. A repair algorithm is the generic name given to any algorithm that is designed to quickly transform infeasible solutions into feasible solutions. Repair algorithms are often used in evolutionary (genetic) algorithms (e.g. Michalewicz and Janikow, 1991; Beasley, 2002); or in constraint satisfaction algorithms (e.g. Minton et al., 1992); or in artificial intelligence approaches (e.g. Chien et al., 1999); and typically systematically replace solution elements that cause constraint conflicts in order to generate a feasible solution. The key similarities and differences between repair algorithms as commonly encountered in the literature and the demolition and reconstruction strategy in our ICH algorithm are:

- the demolition phase (line 13 in figure 1), where we deliberately create infeasibility, has no equivalent in repair algorithms
- the reconstruction phase (lines 15–20 in figure 1) is guided not just by the necessity of restoring feasibility (as in repair algorithms) but also (line 17 in figure 17) chooses the feasible repair that makes the best contribution to the objective function (unlike a number of repair algorithms presented in the literature)

The term "mutation" in an evolutionary (genetic) algorithm is usually taken to denote small (minor) changes that are made to solutions, typically such changes occurring stochastically. The term "macro-mutation" essentially refers to mutations (changes in the solution) that occur on a "larger scale" and as such change the solution significantly more than minor mutations (e.g. Fogel and Angeline, 1998). In this light the demolition phase in ICH, where we deconstruct (demolish) a significant part of the solution, can be regarded as analogous to the devolution process that has been discussed in the context of macro-mutation (e.g. Lattaud, 1998). A key difference here though is that typically macro-mutations do not involve a structured reconstruction phase, unlike our ICH algorithm.

The key to producing good solutions with ICH is the stochastic nature of the demolition and reconstruction strategy. This allows blocks to be repeatedly selected. Hence, a block selected during one reconstruction for a specific processing option, but which contributes little to the global objective, can be selected again during later reconstructions when an alternative processing option may contribute more to the global objective.

4. An example

In this section we illustrate the working of ICH for the OSP for our first test problem, OSP1, involving 2,500 blocks and four processing options, which has been created to represent part

Table 2. Gold mining parameters.

Processing option	Waste	Dump leach	Heap leach	CIP
k	1	2	3	4
Cost (\$/ton) $- c_k$	1.50	3.525	5.775	10.275
Recovery $- p_k$	0	0.45	0.70	0.95

Note: $q_b = 12.5$ tons $\forall b$ and $v = \$9.00$/g are constants.

of a typical gold mine. The number of feasible solutions for OSP1 is $4^{2500} = 1.4 \times 10^{1505}$. Note here that all of the test problems solved in this paper are publicly available from OR-Library (Beasley, 1990, 1996), see the Web page: http://mscmga.ms.ic.ac.uk/jeb/orlib/ospinfo.html.

The mining parameters for this problem are shown in Table 2. Dump Leach processing involves placing blasted rock material on a pad to extract the gold using a cyanide leaching technique. In the Heap Leach process the blasted material is coarsely crushed prior to being placed on the pad. Crushing the ore involves an additional cost, but increases the proportion of gold recovered due to the enhanced surface area exposed to the leaching agent. The vast majority of gold in finely crushed ore can be extracted using an expensive carbon-in-pulp (CIP) milling process. One hundred conditional simulations of z-values were generated by sequential Gaussian simulation (Goovaerts, 1997, pp. 380–392). The simulated z-values (measured in grams per ton) together with the information shown in Table 2 were used to calculate F_S. Note here that all the computational times given in this paper exclude the time required to produce the z-values via simulation.

We consider an implementation (Implementation 10 in Table 1) of the ICH heuristic with $I = 20, \lambda = 0.99, \alpha = 2$ and $t = \$350{,}000$. Figure 2 shows the block processing decisions for a number of iterations, and Table 3 the corresponding objective function and ICH parameter values. Note that although a large proportion of the solution may be demolished there may only be a few changes between the current and reconstructed solutions. For example, during the seventh iteration, 147 block processing decisions are removed from the current solution. However, the reconstructed solution contains only 14 blocks for which the processing decision has changed from the current solution. Computationally we observed that the ratio (local decision changes in the reconstructed solution/number of block decisions demolished) decreases rapidly as the number of iterations increases. As an illustration of this we have that in figure 2, after three iterations, the majority (89.6%) of the block processing decisions correspond to those in final solution, which was reached after 3487 iterations.

5. Computational comparison

In this section we report the results of computational tests carried out to compare the solutions returned by ICH with those returned by a commonly used neighbourhood search technique, simulated annealing (SA) (Kirkpatrick, Gelatt, and Vecchi, 1983). The first problem instance (OSP1) considered here is the same as in the previous section, but the

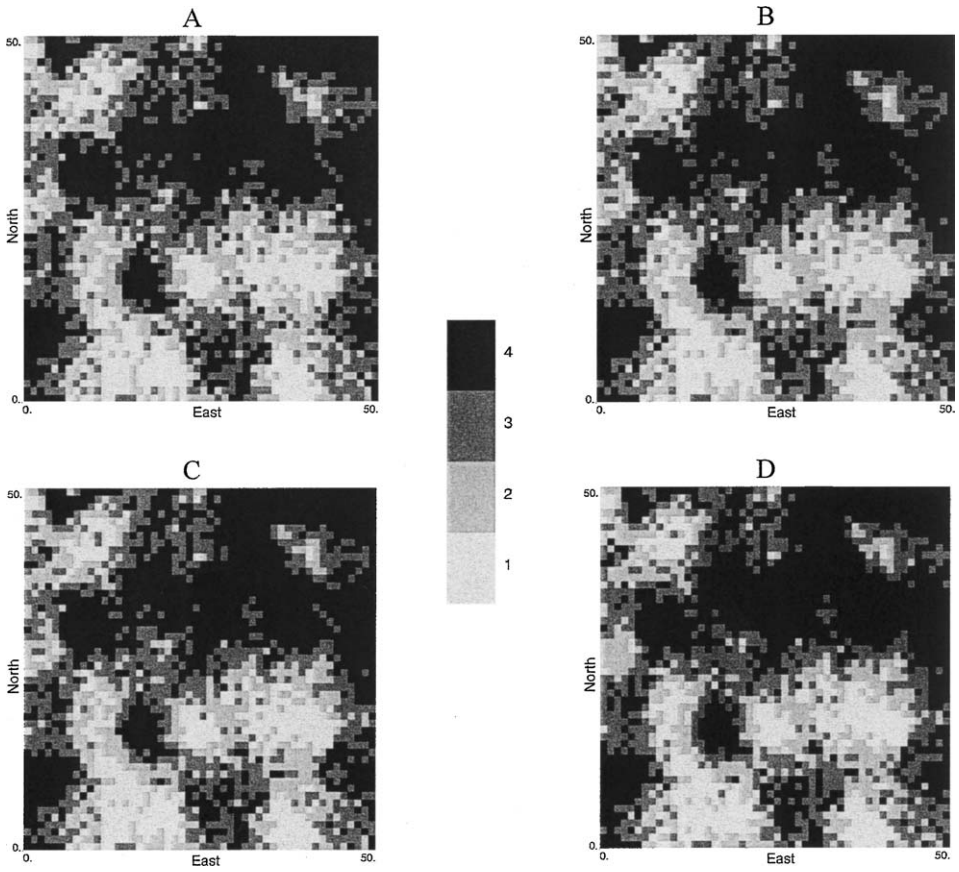

Figure 2. ICH solutions: (A) after one iteration; (B) after two iterations; (C) after three iterations; and (D) after 3487 iterations (final solution).

goal in this section is to generate the efficient frontier. One hundred different values of λ were considered varying between 0.99 and 0.999999. The risk function parameters were fixed at $\alpha = 2$ and $t = \$350{,}000$.

One issue with respect to ICH for the OSP that is worthy of mention here relates to the role of t, the target payoff, when evaluating partial solutions (i.e. solutions for which some blocks do not have processing options decided). It is clear that partial solutions may in many cases, because of their very nature, be unable to reach the target payoff level t. In such cases the objective function evaluation (Eq. (8)) of any partial solution can give excessive weight to the term relating to risk. In order to avoid this we, in evaluating any partial solution $S^{\#}$ (line 17 in figure 1), used a value for target payoff in Eq. (8) equal to t(number of blocks for which a processing option has been decided in $S^{\#}/B$), i.e. we scaled the target payoff value linearly in proportion to the number of blocks for which a processing decision had been made.

Table 3. Objective function and ICH parameter values.

Iteration	$V(S)$	m	y_{bk} changes	∂	Improved solution?
1	−1,125,867	2,500	1,826	−40	Yes
2	−1,111,248	1,747	302	−60	Yes
3	−1,110,696	145	21	−80	Yes
4	−1,110,696	11	2	−79	No
5	−1,110,696	1	1	−78	No
6	−1,095,195	1,006	132	−98	Yes
7	−1,089,628	147	14	−118	Yes
–	–	–	–		–
–	–	–	–		–
–	–	–	–		–
3485	−984,312	2	0	18	No
3486	−984,312	3	0	19	No
3487	−984,312	51	0	20	No

To illustrate the worth of scaling t in this manner figure 3 shows (downside) risk versus expected payoff for ICH after one iteration using both unscaled and scaled t for a number of λ values. Note that solutions generated with unscaled t are restricted to the higher expected payoff parts of this figure. Solutions generated with scaled t cover a larger range of expected

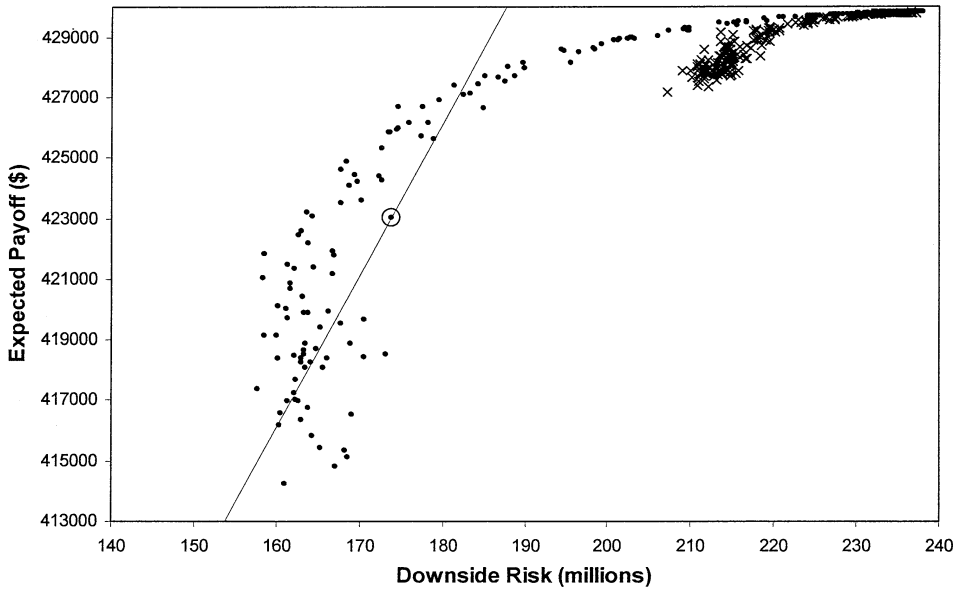

Figure 3. ICH solutions using unscaled t (crosses) and scaled t (dots).

payoffs, and are often financially more efficient than those solutions found using unscaled t. This can be seen for example by considering the indifference line with slope 0.0005 passing through a solution S^* represented by the point (173.9, $423,039) circled in figure 3. This indifference line represents points in payoff-risk space for which the objective function (Eq. (8) with $\lambda = 1/1.0005$) is equal to the objective function value at S^*. Solutions that fall to the left of this indifference line dominate S^* as they represent a better (weighted) tradeoff between risk and payoff than S^*. Conversely, S^* dominates solutions that fall to the right of this indifference line.

Note here that in figure 3 if we examine the relative magnitudes of the values given for downside risk (of the order of hundreds of millions) and expected payoff (of the order of hundreds of thousands) we might at first sight wonder why anyone would ever bother to do any mining at all in this gold mine, since risk seems to dramatically outweigh payoff. The reason for the disparity in values here lies in the definition of downside risk (Eq. (7)) which, since we use $\alpha = 2$, is actually expressed in squared dollars, whereas expected payoff is expressed in dollars.

The implementation strategies numbered 6–8 in Table 1 were used for both ICH and SA. For ICH, I was set to 20 for all λ values. The heuristic (discrete) efficient frontier for ICH using Implementation 8 (ICH-8) is shown in figure 4. This diagram also shows solution paths as solid lines for a number of different λ values. Note that solutions after just one iteration of ICH (i.e. $C = 1$) may be in close proximity to the efficient frontier, but still distant from the final solution found by ICH. This feature is most obvious in figure 4 for expected payoffs greater than $427,000, which correspond to high λ values.

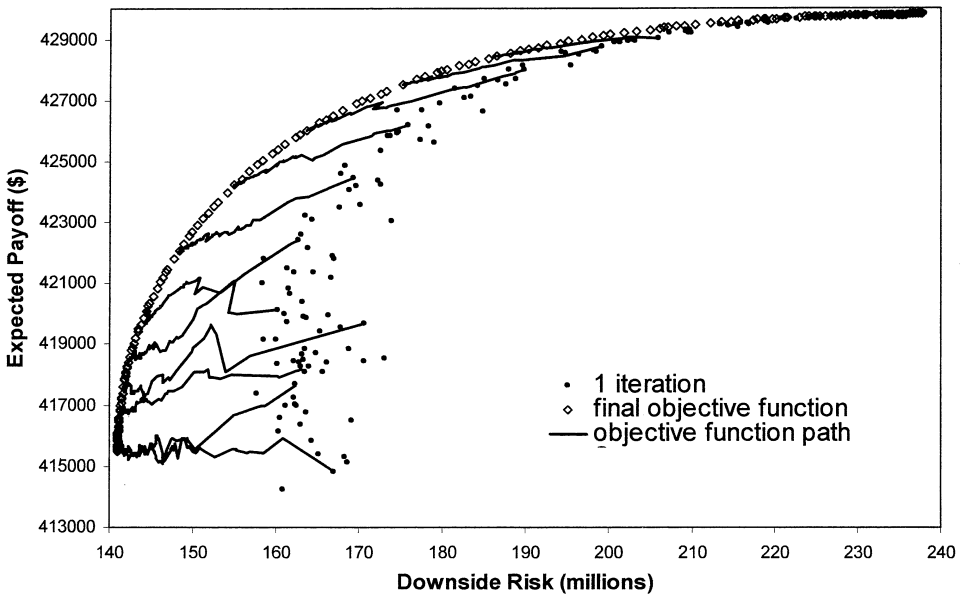

Figure 4. ICH efficient frontier.

To make the comparison between ICH and simulated annealing as fair as possible the initial solution in SA was provided by a CH, simply applying ICH but for only one iteration. Unlike ICH, the perturbation approach in SA was random. In SA, the probability of accepting a deteriorated solution was assumed to be a Boltzmann-like distribution, defined as $e^{-\Delta/T}$, where Δ is the difference in value between the original solution and the deteriorated solution and T is the current temperature. The initial temperature was 0.0001 and this was decreased by a factor of ten after 1000 perturbations had been accepted or 5000 perturbations had been attempted. The low initial temperature avoided significant randomisation of the initial solution provided by the CH. For each λ value SA was stopped if there was no improvement in the solution after three consecutive temperature reductions, or after a maximum of 250,000 iterations.

The results from three ICH and three CH-SA implementations for OSP1 are summarised in Table 4. In this table, the quality of the implementation is related to the sum of the final objective function values for all λ values. This sum is compared in a similar manner as in Eq. (9) with the sum of the maximum objective function values (found over all implementations) for each λ value. Thus, only a heuristic that produced the best solution for all λ values would have a quality value of zero. The lower the quality value in Table 4 the better the efficient frontier for the OSP produced by the heuristic. The quality of efficient frontiers produced by the ICH implementations improves as the maximum number of blocks J that can be demolished during any iteration increases. However, there is also a corresponding increase in computational effort. The CH-SA heuristic performed poorly when more than two local decisions were perturbed at any iteration in SA. ICH-8 produced the best quality efficient frontier and it is clear from Table 4 that it substantially outperforms CH-SA-6, which is itself by far the best of the CH-SA implementations.

To assess the effect of the OSP size on the performance of ICH and CH-SA, a second problem instance (OSP2)—involving 78,000 blocks and the processing options shown in

Table 4. Summary of ICH and CH-SA results.

Implementation[a]	Time (s)	Quality
	OSP1	
ICH-6	63	0.1135
ICH-7	158	0.0017
ICH-8	419	0.0003
CH-SA-6	460	0.0021
CH-SA-7	587	0.2502
CH-SA-8	283	6.9101
	OSP2	
ICH-6	1,513	0.00916
ICH-7	2,904	0.00034
ICH-8	10,752	0.00014
CH-SA-6	8,997	0.00074

[a]Numbers correspond to implementations in Table 1.

Table 2—was also considered. The number of feasible solutions is 4^{78000}. The sizes adopted for OSP1 and OSP2 in this paper were selected to cover the range of OSP sizes that would commonly be encountered in real-life mining situations. For OSP2 the efficient frontier was generated with risk function parameters $\alpha = 2$ and $t = \$4{,}550{,}000$. One hundred different values of λ were considered varying between 0.970874 and 0.999999. Twenty-five conditional simulations of z-values were generated by sequential Gaussian simulation.

The results from three ICH and one CH-SA implementation for OSP2 are summarised in Table 4. As for OSP1, there is a trade-off between speed and quality of the solution when using ICH. The results from ICH-7 and CH-SA-6 indicate that better quality solutions can be generated in a significantly shorter time with the proposed ICH heuristic than with the relay hybrid heuristic CH-SA. As for OSP1, ICH-8 substantially outperforms CH-SA-6.

6. Conclusions

In this paper we presented a greedy iterative constructive heuristic (ICH) that is a variant of other relay hybrid (constructive-neighbourhood search) heuristics. This heuristic was applied to the ore selection problem, which is a discrete decision problem involving uncertainty. We presented computational results on two instances of the ore selection problem (these instances being made publicly available for use by future workers), and compared the solutions returned by our heuristic to those returned by a constructive-simulated annealing relay heuristic. This comparison was carried out both in terms of solution quality and computation time. For the ore selection problems considered, our heuristic outperformed the constructive-simulated annealing hybrid in terms of both solution quality and computational effort.

ICH involves an innovative perturbation strategy that is implemented by repeated partial demolition of the current solution, and reconstruction from the resulting partial solution to provide a new solution in the neighbourhood of the current solution. The key to generating improved new solutions is the stochastic nature of the demolition and reconstruction process.

References

Altenbernd, P. and H. Hansson. (1998). "The Slack Method: A New Method for Static Allocation of Hard Real-Time Tasks." *Real-Time Systems* 15, 103–130.

Beasley, J.E. (1990). "OR-Library: Distributing Test Problems by Electronic Mail." *Journal of the Operational Research Society* 41, 1069–1072.

Beasley, J.E. (1996). Obtaining Test Problems via Internet. *Journal of Global Optimization* 8, 429–433.

Beasley, J.E. (2002). "Population Heuristics." In P.M. Pardalos and M.G.C. Resende (eds.), *Handbook of Applied Optimization*. Oxford: Oxford University Press, pp. 138–157.

Chang, T.-J., N. Meade, J.E. Beasley, and Y.M. Sharaiha. (2000). "Heuristics for Cardinality Constrained Portfolio Optimisation." *Computers and Operations Research* 27, 1271–1302.

Chien, S., G. Rabideau, J. Willis, and T. Mann. (1999). "Automating Planning and Scheduling of Shuttle Payload Operations." *Artificial Intelligence* 114, 239–255.

Fishburn, P.C. (1977). "Mean-Risk Analysis with Risk Associated with Below-Target Returns." *American Economic Review* 67(2), 116–126.

Fogel, D.B. and P.J. Angeline. (1998). Evaluating Alternative Forms of Crossover in Evolutionary Computation on Linear Systems of Equations. In B. Bosacchi, D.B. Fogel, and J.C. Bezdek (eds.), *Applications and Science of Neural Networks, Fuzzy Systems and Evolutionary Computation*, pp. 56–62.

Goovaerts, P. (1997). *Geostatistics for Natural Resources Evaluation*. New York: Oxford University Press.

Kirkpatrick, S., C.D. Gelatt, and M.P. Vecchi. (1983). "Optimization by Simulated Annealing." *Science* 220(4598), 671–680.

Lane, K.F. (1988). *The Economic Definition of Ore Cut-off Grades in Theory and Practice*. London: Mining Journal Books Limited.

Langley, P. (1992). Systematic and Nonsystematic Search Strategies. In *Proceedings of the 1st International Conference on Artificial Intelligence Planning Systems*, pp. 145–152.

Lattaud, C. (1998). "A Macro-Mutation Operator in Genetic Algorithms." In *Proceedings of the 13th European Conference on Artificial Intelligence*, pp. 323–324.

Michalewicz A. and C.Z. Janikow. (1991). "Handling Constraints in Genetic Algorithms." In *Proceedings of the 4th International Conference on Genetic Algorithms*, pp. 151–157.

Minton, S., M.D. Johnston, A.B. Philips, and P. Laird. (1992). "Minimizing Conflicts: A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems." *Artificial Intelligence* 58, 161–205.

Prestwich, S. (2000). "A Hybrid Search Architecture Applied to Hard Random 3-SAT and Low-Autocorrelation Binary Sequences." In *Proceedings of the 6th International Conference on Principles and Practice of Constraint Programming*, pp. 337–352.

Richmond, A.J. (2002). "Applying Four Different Risk Models in Local Ore Selection." *Natural Resources Research* 11(4), 299–314.

Richmond, A.J. (2003). "Financially Efficient Ore Selections Incorporating Grade Uncertainty." *Mathematical Geology* 35(2), 195–215.

Saab, Y.G. and V.B. Rao. (1991). "Combinatorial Optimization by Stochastic-Evolution." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 10(4), 525–535.

Srivastava, R.M. (1987). "Minimum Variance or Maximum Profitability." *CIM Bulletin* 80(901), 63–68.

Talbi, E.G. (2002). "A Taxonomy of Hybrid Metaheuristics." *Journal of Heuristics* 8(5), 541–564.