Heuristic Strategies for Solving Complex Interacting Stockpile Blending Problem with Chance Constraints

Yue Xie
Optimisation and Logistics
School of Computer Science
The University of Adelaide
Adelaide, SA, Australia

Aneta Neumann
Optimisation and Logistics
School of Computer Science
The University of Adelaide
Adelaide, SA, Australia

Frank Neumann
Optimisation and Logistics
School of Computer Science
The University of Adelaide
Adelaide, SA, Australia

ABSTRACT

The stockpile blending problem seeks to determine how many tonnages of ore that stockpiles provide and to which parcels. This scheduling model maximizes the volume of valuable material in the final production subject to resource capacities, constraints for mill machines, and customer requirements. Motivated by the uncertainty in the geologic input date which affects optimization, we consider the stockpile blending problem with uncertainty in material grades. A non-linear continuous optimization model developed here to integrate uncertain variables to optimization. We introduce chance constraints that are used to guarantee the constraint is violated with a small probability to tackle the stochastic material grades. We investigate a well-known approach in this paper, which is used to solve optimization problems over continuous space, namely the differential evolution (DE) algorithm. In the experiment section, we compare the performance of the algorithm with the deterministic model and three chance constraint models by using a synthetic benchmark. We also evaluate the effectiveness of different chance constraints.

CCS CONCEPTS

• **Applied computing** → **Supply chain management**; • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Stochastic mine planning, stochastic optimization, Differential Evolution, chance-constrained optimization

ACM Reference Format:
Yue Xie, Aneta Neumann, and Frank Neumann. 2021. Heuristic Strategies for Solving Complex Interacting Stockpile Blending Problem with Chance Constraints. In *2021 Genetic and Evolutionary Computation Conference (GECCO '21), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3449639.3459382

ACM ISBN 978-1-4503-8350-9/21/07.
https://doi.org/10.1145/3449639.3459382

1 INTRODUCTION

Mining is the extraction of economically valuable minerals or materials from the earth. This has raised the importance of the production scheduling process due to its significant role in the profitability and efficiency of any mining operation. Mine production scheduling determines the time of excavation and the destination of each block in open pits or underground and deal with a set of operational and physical constraints [30]. The task of the mine production schedule is to generate a mining sequence and ensure the product meets the blending resource constraints. Mine production scheduling problem [18] is a well-study mining engineering problem, and it has received much attention in past decades [22] from both engineering and research.

The production scheduling problem consists of generating a plan to guide mining of selected parts of the ore body that ensure delivery of the budgeted tonnes and grade of the mineral raw material to the mill in the period under consideration. The objective of the problem is to maximize the net present value (NPV) of the mining operation by determining the extraction time of each block of ore in a deposit and the destination to which this block must be sent. This problem is commonly formulated as a Mixed-Integer Program (MIP) with binary variables [5, 10, 34]. However, since it becomes a challenge for the MIP when the problem deals with the blending resource constraints. Lipovetzky et al. [12] introduce a combined MIP for a mine planning problem, which devises a heuristic objective function in the MIP and can improve the resulting search space for the planner. Samavati et al. [29] propose a heuristic approach that combines local branching with a new adaptive branching scheme to tackle the production scheduling problem in open-pit mining.

Stockpiles are essential components in the supply chain of the mining industry, and play a significant role in the real-world mining engineering problem. Jupp et al. [11] introduce that there are four different reasons for stockpiling before material processing: buffering, blending, storing, and grade separation. In open-pit mine production scheduling problem, stockpiles are used for blending different grades of material from the mine or keeping low-grade ore for possible future processing [18, 27]. Robinson [28] discusses the benefits of storing material in stockpiles to control the grade of material. Akaike and Dagdelen [1] propose a model for mine planning considering a stockpile. Recently, Rezakhah and Moreno [26] use a linear-integer model to approximate the open-pit mine production scheduling with stockpiling problem which forces the stockpile to have an average grade above a specific limit. Some researchers present nonlinear-integer models to solve open pit mine production scheduling with stockpiling problems. Tabesh et al. [33] propose a nonlinear model of stockpiles to optimize a comprehensive open-pit mine plan but not give any results. Bley et al. [5] propose a nonlinear model for mine production planning, however, they only consider one stockpile. In this kind of problem, the researchers assume that the material mixed homogeneously in the stockpile, but this problem is still difficult to be solved.

In this paper, we study the stockpile blending problem which is an important component of the mine production scheduling problem. This problem is challenging to address in terms of blending material from stockpiles to match the demands of downstream customers. We define the stockpile blending problem as a combinatorial optimization problem that aims to maximal the tonnage of valuable material from all parcels by finding the percentage that each stockpile provides for the parcels in the whole schedule. Furthermore, the blend strategy should respond to the mining schedule and the market plan where the mine schedule provides the material mining and sending to corresponding stockpiles in each period, and the market plan provides the customer requirement.

Solving the stockpile blending problem in mining optimally is critical, because it determines the cash flows that can add more than several hundred million dollars in magnitude, and at the same time it is based on an uncertain supply of mineralized materials for the resource available in the mine. This uncertainty is acknowledged in the related technical literature to be the major reason for not meeting production expectations [2, 4]. Given its substantial impact on the financial outcome of mining operations, this paper focuses on dealing with the uncertainty in metal content within a mineral deposit being mined.

For the stochastic variables of the stockpile blending problem, we introduce chance-constrained programming here to tackle the uncertainty of material grades. Chance-constrained optimization problems [6, 17] whose resulting decision ensures the probability of complying with the constraints and the probability level of being feasible to have received significant attention in the literature. Chance-constraint programming has been widely applied in different disciplines for optimization under uncertainty [9, 20, 35]. For example, chance-constraint programming has been applied in analogue integrated circuit design [15], mechanical engineering [16], and other disciplines [3, 13, 24]. However, so far, chance-constraint programming has received little attention in the evolutionary computation literature [14, 36, 37].

It is difficult for MIP to tackle such a continuous optimization problem with nonlinear constraints. To address this challenge, this paper proposes a decision variable normalized operator to ensure any created variables of one parcel's sum is equal to one. Moreover, in order to tackle the strict tonnes concentrate constraint, we introduce a duration repair operator that searches the duration of parcels in available intervals to ensure the parcels meet the tonnes concentrate constraint. In this paper, we consider the stockpile blending problem with chance constraints, the objective is the same as the deterministic setting of the problem and its subjects to the condition that the probability of constraints that violated the given bound is less than a given threshold.

Furthermore, this paper investigates how to solve the stockpile blending problem by using a well-known evolutionary algorithm, the Differential Evolution (DE) algorithm. Recently, evolutionary algorithms have received much attention in solving large-scale optimization problems and multi-dimensions problem. The DE algorithm is a simple and effective evolutionary algorithm used to solve global optimization problems in a continuous domain [19, 23]. The DE algorithm was first proposed in [25, 31, 32], after then, DE and its variants have been successfully applied to solve numerous real-world problems from diverse domains of science and engineering [8, 19]. Recently, survey papers [7, 8, 21] provide an up-to-date view on DE algorithm and discuss its various modifications, improvements and uses.

The rest of the paper is organized as follows. In the next section, we present the model of the stockpile blending problem and a decision variable normalized operator for the continuous decision variables as well as a duration repair operator. After that, the chance constraints model and the surrogate functions of the chance constraints are presented in Section 3. Following, we describe the approach we used to solve the problem and the fitness function of the algorithm. We set up experiments and investigate the performance of the different fitness functions in Section 5. We conclude with Section 6.

2 DETERMINISTIC MODEL

In this section, we present nonlinear formulations of the stockpiles blending problem with a deterministic setting. In reality, some processes such as the chemical process in the concentrate production progress are highly complex to model because it is influenced by many factors, some of which include the mineralogy of the ore, particle size of milled material, temperature, and chemical reactants available in the process. The information of these variables was not available to us, therefore within this study to recovery factors of all materials from the chemical processing stage and the copper percentage within the produced copper concentrate is assumed to be constant throughout the stockpiles blending and production schedule.

We first introduces notation in table 1, and then provide the math. We use the term "material" to include ore, i.e., rock that contains sufficient minerals including metals that can be economically extracted and to include waste.

$$Obj: \max \sum_{p \in \mathcal{P}} c_p = max \sum_{p \in \mathcal{P}} \left(w_p g_p^{Cu} r_p^{Cu} \right) \tag{1}$$

$$s.t. \sum_{\sum_{m=1}^{m-1} N_m + 1 \leq p \leq \sum_{m=1}^{m} N_m} t_p \leq D^m \tag{2}$$

$$\sum_{s \in \mathcal{S}} x_{ps} = 1 \qquad \forall p \in \mathcal{P} \tag{3}$$

$$r_p^{Cu} = \mu_1^{Cu} \frac{g_p^{Cu}}{g_p^S} + \mu_2^{Cu} \qquad \forall p \in \mathcal{P} \tag{4}$$

$$g_p^o = \sum_{s \in \mathcal{S}} x_{ps} \tilde{g}_{ps}^o \qquad \forall p \in \mathcal{P} \tag{5}$$

$$\begin{aligned} w_p =& \delta t_p [\tilde{\phi} + (\phi^{Au} \log g_p^{Au}) + (\phi^U \log g_p^U) \\ &- (\phi^{Fe} \log g_p^{Fe}) + (\phi^{Cu} \log g_p^{Cu})] \end{aligned} \tag{6}$$

Table 1: Notation

Indices and sets:	
$s \in \mathcal{S}$	stockpiles; $1, \ldots, S$
$p \in \mathcal{P}$	parcels; $1, \ldots, P$
o	material; material; $\{Cu, Ag, Fe, Au, U, Fl, S\}$
$m \in \mathcal{M}$	month; $1, \ldots, m$
Decision variables:	
x_{ps}	fraction of parcel p claimed from stockpile s
t_p	produce time (duration) for parcel p
w_p:	tonnage of parcel p
θ_{ps}:	tonnage stores in stockpile s after providing material to parcel p
c_p:	copper tonne in parcel p
g_p^o:	grade of material o in parcel p
$\tilde{g}_{ps}^o$:	grade of material o in stockpile s when proving parcel p
k_p:	tonne concentrate of parcel p
r_p^{Cu}:	Cu recovery of parcel p
r_p^{Fl}:	Fl recovery of parcel p
Parameters:	
T_p^m:	binary parameter, if $T_p^m = 1$, parcel p is the parcel need to prepare in month m, if $T_p^m = 0$ otherwise
δ:	discount factor for time period
$\tilde{\phi}$:	factor in chemical processing stage
ϕ^{Au}:	factor of Au in chemical processing stage
ϕ^{U}:	factor of U in chemical processing stage
ϕ^{Fe}:	factor of Fe in chemical processing stage
ϕ^{Cu}:	factor of Cu in chemical processing stage
(γ_1, γ_2) :	factor of copper percentage within the produced copper concentrate
μ^{Fl}:	factor of Fl recovery
μ^{U} :	factor of U recovery
(μ_1^{Cu}, μ_2^{Cu}) :	factor of copper recovery
D^m:	duration of month m
H_s^m:	tonnage of material hauled to stockpile s in month m
G_s^{om}:	grade of material o that shipping to the stockpile s in month m
K_p:	expected tonne concentrate of parcel p
R_p^{Fl} :	upper threshold of Fluorine recovery of parcel p
Cu_p :	lower threshold of Copper grade of parcel p
N_m:	number of planning parcels in month m

$$k_p = \frac{c_p}{\gamma_1 \frac{g_p^{Cu}}{g_p^S} + \gamma_2} \tag{7}$$

$$\tilde{g}_{ps}^o = \begin{cases} \frac{\tilde{g}_{(p-1)s}^o \cdot \theta(p-1)s + G_s^{om} \cdot H_s^m}{\theta(p-1)s + H_s^m} & if\, T_p^m = 1 \\ \tilde{g}_{(p-1)s}^o & otherwise \end{cases} \tag{8}$$

$$\theta_{ps} = \begin{cases} \theta_{(p-1)s} + H_s^m - x_{ps} \cdot w_p & if\, T_p^m = 1 \\ \theta_{(p-1)s} - x_{ps} \cdot w_p & otherwise \end{cases} \tag{9}$$

$$g_p^{Cu} \geq Cu_p \qquad \forall p \in \mathcal{P} \tag{10}$$

$$(K_p - 1) \leq k_p \leq (K_p + 1) \qquad \forall p \in \mathcal{P} \tag{11}$$

$$\mu^{Fl} g_p^{Fl} \leq R_p^{Fl} \qquad \forall p \in \mathcal{P} \tag{12}$$

The objective function is the sum of Cu tonne in all planning parcels, which is obtained by the tonnage of parcels multiply the copper grade of parcels, multiple the copper recovery of parcels. The first constraint (2) forces the sum of the duration of the parcels that planned into the same month equal to the available duration of this month.

Constraint (3) ensures that the sum of the fractions of the same parcel equal to 1. Function (4) expresses the simplified calculation of copper recovery of each parcel, and function (5) calculates the material grades of parcels. Function (6) express the simplified calculation of parcel tonne which is a component in objection function. Function (7) shows the simplify version of how to calculate the tonne concentrate of parcels.

Algorithm 1: Decision variables normalized approach

Input: Decision vector $X_j = \{x_{1j}, x_{2j} .., x_{Ij}\}$
$a = \sum_{i \in I} x_{ij}$;
for $i = 1$ *to* I **do**
  $x_{ij} = \frac{x_{ij}}{a}$;
return the normalized decision variables.

Constraint (8) enforces material grade balance for stockpiles when providing material to parcels. When the parcels which is the first parcel within a month claimed material from stockpiles, the material grade of each stockpile is equal to the sum of material tonne store in the stockpile after preparing the last parcel and the material tonne in the ore that sends from mine, then divides by the sum of tonnage store in the stockpile after preparing last parcel and the tonnage of ore that send from mine. Constraint (9) enforces inventory balance when providing material to parcels, ensuring that the amount of material in the stockpiles in the period after provided material to the first parcel in each month equal to that of the last parcel plus anything that was added and minus anything sent to the current parcel from the stockpile. Furthermore, the material store in each stockpile when providing other parcels is equal to that of the last parcel minus anything sent to the parcel from the stockpile. Constraint (10) forces the copper grade of each parcel to be less than or equal to the pre-given lower threshold of Cu grade of each parcel. Constraint (11) forces the value of tonne concentrate of each parcel is not more or less than the expected tonne concentrate by one. Constraint (12) ensures the Fl recovery of each parcel less than the bound given in advance.

The stockpile blending problem is a non-linear optimization problem in continuous search space. To tackle the constraint (3), we introduce a decision variables normalized approach (cf. Algorithm 1) to force the solution match the constraint. This approach first calculates the sum of the variables of each parcel separately, then each variable of the same parcel is divided by the corresponding sum. It shows the significant importance of applying this approach to a generated decision vector to meet the constraint.

For the tight constraint (11), it is hard to generate a feasible solution that forces the real tonne concentrate of a parcel no more or less one than the given threshold. As shown in equation (7), the value of toner concentrate of each parcel is related to the duration of this parcel and material grades of this parcel. Meanwhile, referring to equation (5), metal grades of the parcel are directly calculated by decision variables. Therefore, with the determined decision variables of a parcel, the real tonnes concentrate on this parcel is affected by the duration of this parcel. Now, we present a duration repair operator (cf. Algorithm 2) which uses a binary search process to convert an infeasible solution into a solution without violating constraint (11).

Since the time complexity of the binary search is $\log n$ where n denotes the length of the search space in the beginning. In our

Algorithm 2: Duration repair operator

Input: $X \in (0,1)^{I \cdot J}$, $i \in \{1,..,I\}$, $j \in \{1,..,J\}$; parameter ζ; available duration D
Output: parcel duration: $d \in \{0, D\}$
initialization: $\underline{d} = 0, \overline{d} = D, d \in \{0, D\}$, $k = \zeta \cdot d$ **while** $d \in \{0, D\}$ *and* $k \notin \{K-1, K+1\}$ **do**
  if $k > K+1$ **then**
    $d := (d + \underline{d})/2$;
    $k := \zeta \cdot d$;
    if $k > K+1$ **then**
      $\overline{d} = d$;
    else
      $\underline{d} = d$;
  else if $k < K-1$ **then**
    $d := (d + \overline{d})/2$
  $k := \zeta \cdot d$;
  if $k > K+1$ **then**
    $\overline{d} := d$;
  else
    $\underline{d} := d$
return *the duration corresponding to solution X*

problem, the duration of each parcel can not exceed the total available duration of the month. The run-time of the duration repair operator for one parcel is $\log d$ in the worst case where d denotes the total available duration of the current month.

3 MODEL WITH CHANCE CONSTRAINTS

In this paper, we are the first to discuss the effeteness of stochastic material grades on the objective value of the stockpile blending problem. Moreover, due to the complexity of the problem, we reformulated the constraints (10) and (12) to chance constraints respectively. Chance constrained programming is a competitive tool for solving optimization problems under uncertainty. Its main feature is that the resulting decision ensures the probability of complying with constraints, i.e. the probability of being feasible. Thus, using chance-constrained programming the relationship between profitability and reliability can be quantified.

3.1 The formulation of the chance constraints

First, we define additional notation as follow.

α_{Cu}:	probability threshold of copper grade chance constraint
α_{Fl}:	probability threshold of Fl recovery chance constraint

Then, the new chance constraints are:

$$Pr\{g_p^{Cu} \geq Cu_P\} \geq \alpha_{Cu}, \tag{13}$$

$$Pr\{\mu^{Fl} g_p^{Fl} \leq R_p^{Fl}\} \geq \alpha_{Fl}. \tag{14}$$

Constraints (13), (14) force the probability of guarantying the given bound are greater than or equal to the corresponding given threshold.

In this paper, we use Chebyshev's inequality to construct the available surrogate that translates to a guarantee on the feasibility of the chance constraint imposed by the inequalities. Firstly, we use Chebyshev's inequality to reformulate the chance constraints. The inequality has utility for being applied to any probability distribution with known expectation and variance. Therefore, we assume the stochastic material grades discussed in this paper are all estimated with given expected values and corresponding variances. Note that Chebyshev's inequality automatically yields a two-sided tail bound, there is a one-sided version of Chebyshev's inequality named Cantelli's inequality.

Theorem 3.1 (Cantelli's inequality). *Let X be a random variable with $Var[X] > 0$. Then for all $\lambda > 0$,*

$$P_r\{X \geq E[X] + \lambda\sqrt{Var[X]}\} \leq \frac{1}{1+\lambda^2}. \tag{15}$$

$$P_r\{X \leq E[X] - \lambda\sqrt{Var[X]}\} \leq \frac{1}{1+\lambda^2}. \tag{16}$$

We assume the material grades in the ore that hauled to stockpiles are independent of each other, each grade corresponding expectation a_s^{om} and variance σ_s^{2om}. Therefore, the expected material grades of stockpiles in the periods when providing material to the first parcel in months can be denoted as

$$E(\tilde{g}_{ps}^o) = \frac{\tilde{g}_{(p-1)s}^o \cdot \theta_{(p-1)s} + a_s^{om} \cdot H_s^m}{\theta(p-1)s + H_s^m},$$

$$E(\tilde{g}_{ps}^o) = E(\tilde{g}_{(p-1)s}^o)$$

when providing material to other parcels. Furthermore, the variance of the material grades are

$$\mathrm{Var}(^o_{ps}) = \left(\frac{\theta_{(p-1)s}}{\theta(p-1)s+H_s^m}\right)^2 Var(\tilde{g}_{(p-1)s}^o) + \left(\frac{H_s^m}{\theta(p-1)s+H_s^m}\right)^2 \sigma_s^{2om},$$

$$Var(\tilde{g}_{ps}^o) = Var(\tilde{g}_{(p-1)s}^o)$$

for different parcels respectively.

Let $g_p^{Cu} = \sum_{s \in \mathcal{S}} x_{ps} \tilde{g}_{ps}^{Cu}$ be the copper grade of parcel p of a given solution $X = \{x_{p1}, .., x_{ps}, .., x_{p\mathrm{S}}\}$, and

$$E[g_p^{Cu}] = \sum_{s \in \mathcal{S}} x_{ps} E(\tilde{g}_{ps}^{Cu})$$

denotes the expected copper grade of parcel p of the solution derived by linearity of expectation,

$$Var[g_p^{Cu}] = \sum_{s \in \mathcal{S}} (x_{ps})^2 Var(\tilde{g}_{ps}^{Cu})$$

denotes the variance of copper grade of parcel p. To match the expression of the Cantelli's inequality (16), we set

$$Cu_p = E[g_p^{Cu}] - \lambda\sqrt{Var[g_p^{Cu}]}$$

and have

$$\lambda = \frac{E[g_p^{Cu}] - Cu_p}{\sqrt{Var[g_p^{Cu}]}}$$

for each parcel, then we have a formulation to calculate the upper bound of the chance constraint (13) as follows.

$$Pr\{g_p^{Cu} \leq Cu_p\} \leq \frac{Var[g_p^{Cu}]}{Var[g_p^{Cu}] + (E[g_p^{Cu}] - Cu_p)^2} \leq (1 - \alpha_{Cu}) \tag{17}$$

Furthermore, let $r_p^{Fl} = \mu^{Fl} \sum_{s \in \mathcal{S}} x_{ps} \tilde{g}_{ps}^{Fl}$ be the FL recovery of parcel p. Let

$$E[r_p^{Fl}] = \mu^{Fl} \sum_{s \in \mathcal{S}} x_{ps} E(\tilde{g}_{ps}^{Fl})$$

denotes the expectation of Fl recovery, and

$$Var[r_p^{Fl}] = \sum_{s \in \mathcal{S}} (\mu^{Fl} x_{ps})^2 Var(\tilde{g}_{ps}^{Fl}),$$

is the variance of FL recovery of parcel p with solution $X = \{x_{p1}, .., x_{ps}, .., x_{p\mathcal{S}}\}$.

To match the expression of the Cantelli's inequality (15), we set

$$R_p^{Fl} = \mu^{Fl} E[r_p^{Fl}] + \lambda \sqrt{Var[r_p^{Fl}]}$$

and have

$$\lambda = \frac{R_p^{Fl} - \mu^{Fl} E[g_p^{Fl}]}{\sqrt{Var[g_p^{Fl}]}}$$

for each parcel, then we have a formulation to calculate the upper bound of the chance constraint (14) as follows.

$$Pr\{\mu^{Fl} g_p^{Fl} \geq R_p^{Fl}\} \leq \frac{Var[g_p^{Fl}]}{Var[g_p^{Fl}] + (R_p^{Fl} - \mu^{Fl} E[g_p^{Fl}])^2} \leq (1 - \alpha_{Fl}) \tag{18}$$

Now, we obtain the surrogate functions of the chance constraints. In the next section, we present the approach for solving the stockpile blending problem with chance constraints.

4 APPROACHES FOR THE STOCKPILE BLENDING PROBLEM WITH CHANCE CONSTRAINTS

In this section, we present the fitness function of the differential evolution (DE) algorithm which has been proved successfully used in solving the optimization problem in continuous space.

4.1 Fitness function of the problem with deterministic setting

We start by designing a fitness function for the deterministic setting model that can be used in DE. The fitness function f for the approach needs to take all constraints into account. The fitness function of a solution X is defined as follows.

$$f(X) = (u(X), v(X), w(X), q(X), g(X), O(X)) \tag{19}$$

$$\begin{aligned}
u(X) &= \sum_{p \in \mathcal{P}} \max\{|K_p - k_p|, 1\} \\
v(X) &= \max\{\sum_{p \in \mathcal{P}} t_p - D^m, 0\} \\
w(X) &= \min\{\sum_{p \in \mathcal{P}} \sum_{s \in \mathcal{S}} \theta_{ps}, 0\} \\
q(X) &= \sum_{p \in \mathcal{P}} \max\{Cu_p - g_p^{Cu}, 0\} \\
g(X) &= \sum_{p \in \mathcal{P}} \max\{r_p^{Fl} - R_p^{Fl}, 0\} \\
O(X) &= \sum_{p \in \mathcal{P}} c_p.
\end{aligned}$$

In this fitness function, the components u, v, q, g need to be minimize while w and O maximized, and we optimize f in lexicographic order. For the stockpile blending problem, any infeasible solution can at least violate one of the above constraints. Then, among solutions that meet all constraints, we aim to maximize the objective function. Formally, we have

$$\begin{aligned}
&f(X) \geq f(Y) \\
\textbf{iff}\ &u(X) < u(Y)\ \textit{or} \\
&u(X) = u(Y) \wedge v(X) < v(Y)\ \textit{or} \\
&\{u, v\}\text{are equal} \wedge w(X) > w(Y)\ \textit{or} \\
&\{u, v, w\}\text{are equal} \wedge q(X) < q(Y)\ \textit{or} \\
&\{u, v, w, q\}\text{are equal} \wedge g(X) < g(Y) \textit{or} \\
&\{u, v, w, q, g\}\text{ are equal} \wedge O(X) > O(Y),
\end{aligned}$$

When comparing two solutions, the feasible solution is preferred in a comparison between an infeasible and a feasible solution. Between two infeasible solutions that violated the same constraint, the one with a lower degree of constraint violation is preferred.

4.2 Fitness function of the problem with chance constraints

Now, we design the fitness function for the stockpile blending problem with chance constraints. In this paper, we investigate the effectiveness of chance constraints on the objective value. We first reformulate the components q and g of the function (19) with chance constraints (17 and 18) as follow,

$$q'(X) = \sum_{p \in \mathcal{P}} max \left\{P_r\{g_p^{Cu} \leq Cu_p\} - (1 - \alpha_{Cu}), 0\right\} \tag{20}$$

$$g'(X) = \sum_{p \in \mathcal{P}} max \left\{P_r\{\mu^{Fl} g_p^{Fl} \geq R_p^{Fl}\} - (1 - \alpha_{Fl}), 0\right\} \tag{21}$$

where q' and g' need to be minimized.

To distinguish the influence of each chance constraint, we design three fitness functions where the two functions consider the chance constraints separately, and the other one uses the combination of

Algorithm 3: Differential evolution algorithm

$t \leftarrow 1$, initialize $\mathbf{P}^t = \{X_1^t, .., X_{NP}^t\}$ randomly ;
while *stopping criterion not met* **do**
  for $i \in \{1, .., NP\}$ **do**
    $R \leftarrow$ A set of randomly selected indices from $\{1, .., NP\} \setminus \{i\}$;
    $V_i^t \leftarrow$ mutation (P^t, R, F) ;
    $j_{rand} \leftarrow$ A randomly selected number from $\{1, .., n\}$;
    $U_i^t \leftarrow$ crossover$(X_i^t, V_i^t, C_r, j_{rand})$;
  for $i \in \{1, .., NP\}$ **do**
    if $f(U_i^t) \geq f(X_i^t)$ **then**
      $X_i^{t+1} \leftarrow U_i^t$;
    else
      $X_i^{t+1} \leftarrow X_i^t$;
  $t \leftarrow t + 1$;

components.

$$f'(X) = \big(u(X), v(X), w(X), q'(X), g(X), O(X)\big) \tag{22}$$

$$f''(X) = \big(u(X), v(X), w(X), q(X), g'(X), O(X)\big) \tag{23}$$

$$f'''(X) = \big(u(X), v(X), w(X), q'(X), g'(X), O(X)\big) \tag{24}$$

4.3 Differential evolution algorithm

Algorithm (3) shows the overall procedure of the basic DE algorithm. DE is usually initialized by generating a population of NP individuals. For each $i \in \{1, .., NP\}$, X_i^t is the i-th individual in the population $\mathbf{P}^t$. Each individual represents a d-dimensional solution of a problem. For each $j \in \{1, .., d\}$, X_{ij}^t is the j-th element of X_i^t.

After the initialization of $\mathbf{P}^t$, the following steps are repeatedly performed until a termination condition is satisfied. For each X_i^t, the scale factor $F > 0$ which controls the magnitude of the mutation, and the crossover rate $Cr \in [0, 1]$ which controls the number of elements inherited from X_i^t to a trail vector U_i^t are constants and given in advance.

A set of parent indices $R = \{r_1, r_2\}$ are randomly selected from $\{1, .., n\}\{i\}$ such that they differ from each other. For each X_i^t, a mutant vector V_i^t is generated by applying a mutation to X_{r1}^t, X_{r2}^t. There are many mutation strategies that have been proposed in the literature [8]. Here, we use the $DE/target-to-best/1$ strategy shown as follows, which is one of the most efficient strategies.

$$V_i^t = X_i^t + F(X_{best}^t - X_i^t) + F(X_{r1}^t - X_{r2}^t), \tag{25}$$

where X_{best}^t denotes the best individual in the current population.

After the mutant vector V_i^t has been generated for each X_i^t, a trail vector U_i^t is generated by applying crossover to X_i^t and V_i^t. The scheme of the crossover can be outlined as

$$U_{ij}^t = \begin{cases} V_{ij}^t & if\,(rand_{i,j}[0,1] \leq Cr \; or \; j = j_{rand}) \\ X_{ij}^t & otherwise \end{cases} \tag{26}$$

where $rand_{i,j}[0, 1]$ is a uniformly distributed random number, which is called a new for each j-th element of the i-th parameter vector. $j_{rand} \in \{1, .., n\}$ is a randomly chosen index, which ensures that U_i^t gets at least one element from V_i^t. It is instantiated once for each vector per generation.

Table 2: General information about the ore and processing parameters

Description	Values or Value range
Number of parcel	stockpiles; $\{3, 4, 5\}$
Number of Stockpile	7
Duration of month	29, 30, 31
Discount factor for time period (δ)	0.98
Factor in chemical processing stage ($\tilde{\phi}$)	$[1000, 2000]$
Factor of Au in chemical processing stage (ϕ^{Au})	$[200, 300]$
Factor of U in chemical processing stage (ϕ^{U})	$[300, 400]$
Factor of Fe in chemical processing stage (ϕ^{Fe})	$[560000, 570000]$
Factor of Cu in chemical processing stage (ϕ^{Cu})	$[6000000, 7000000]$
Factor of copper percentage within the produced copper concentrate (γ_1, γ_2)	$([5, 10], [30, 40])$
Factor of Fl recovery (μ^{Fl})	$[0.05, 0.15]$
Factor of U recovery (μ^{U})	$[0.5, 0.9]$
Factor of copper recovery (μ_1^{Cu}, μ_2^{Cu})	$([1.5, 3.5], [0, 10])$
Tonnage of material hauled to stockpile	$[5000, 1000000]$
Cu grade	$[0.05, 2.5]$
Ag grade	$[1.0, 4.0]$
Fe grade	$[10.0, 30.0]$
Au grade	$[0.3, 2.0]$
U grade	$[30.0, 400.0]$
Fl grade	$[1200, 4500]$
S grade	$[0.15, 1.0]$
Expected tonne concentrate of parcel (K_p)	$[10000, \infty]$
Threshold of Fluorine recovery of parcel (R_p^{Fl})	$[1300, 1500]$
Threshold of Copper grade of parcel (Cu_p)	$[0.5, 1.5]$

After the trial vector, U_i^t has been generated for each parent individual, the next step called selection which determines whether the target or the trailing vector survives to the next generation. The selection operation is described as

$$X_i^{t+1} = \begin{cases} U_i^t & iff(U_i^t) \geq f(X_i^t) \\ X_{ij}^t & otherwise \end{cases} \tag{27}$$

according to the fitness function.

5 EXPERIMENTAL INVESTIGATION

In this section, we examine the solution quality associated with different fitness functions. Due to the business security, we are not able to compare the performance of the DE algorithm in a real-data instance. Therefore, we first design the instances used in this paper. Afterward, we compare the results obtained by using different fitness functions and examine the performance of the algorithm.

5.1 Experimental Setup

Table 2 lists the possible intervals of the tonnage of ore and material grades of the ore shipping from mine to stockpiles used for performance analysis and some of the process parameters.

The three instances we used in this paper is created by randomly generated value of parameters (see Table 2), we attach the parameters of these instance in the appendix. Here, we consider the case that material grades conform to the Normal distribution, the expected values of grades are randomly selected from the value range, and the deviation of material grades are set equal to 0.01 multiply the expected value and let $\alpha_{Cu} = \{0.999, 0.99, 0.9\}$ and $\alpha_{Fl} = \{0.999, 0.99, 0.9\}$. Base on this arrangement, we compare the performance of the DE algorithm with fitness functions (Eq. 19, 22, 23, 24) on the stockpile blending problem.

We investigate the performance of the DE algorithms with different fitness functions described in Section 4 and provide the results from 30 independent runs with 10000 generation and 10 population for all instances. For a closer look, we report the average, best and worst solutions obtained by the algorithm in corresponding columns. We also evaluate the algorithm by success rate which is the percentage of success for the algorithm in obtaining valid solutions out of 30 runs.

5.2 Experimental Results

We benchmark our approach with the combinations from the experimental setting described above. All experiments were performed using Java of version 11.0.1 and carried out on a MacBook with a 2.3GHz Intel Core i5 CPU.

Table 3 lists the results for the three instances with using fitness function (19, 22) and (23) separately, where they consider one of the chance constraints. Figure 1 shows the how the one of the chance-constrained bound $(\alpha_{Cu}, \alpha_{Fl})$ affects the solution quality. The bars in the graphs are corresponding to the solutions of instances combining with the probability of chance constraint respectively, and the three bars in each group corresponding to the value of uncertainty $\{0.999, 0.99, 0.9\}$. Among others, we observe that results obtained by applying the fitness function (22) are significantly affected by the value of uncertainty denoted by α_{Cu}. The results show an increasing trend as the value of α_{Cu} decrease. However, by observing the bars in *Fl chance constraint* group, the value of α_{Fl} does not influence the result when using Fl chance constraint in the fitness function.

As can be seen from Table 3, the success rate shows significantly different results between using Cu chance constraint and Fl chance constraint for instance 1 and 2. When the probability of Cu grade chance constraint is tight such as 0.999, the DE algorithm can not generate a pure feasible population in the last generation. While the probability of Fl recovery chance constraint does not influence the success rate of the algorithm. However, for instance, 3, which has four parcels into consideration and is the most complex instance in our study, the DE algorithm fails to obtain a feasible population in the last generation when the value of probability of Fl recovery chance constraint is 0.999.

Table 4 lists the results obtained by considering two chance constraints, the fitness function (24). For each instance, we investigate different parameters setting together with the different requirement on the chance constraints determined by α_{Cu} and α_{Fl}. The results list in the columns with the same α_{Cu} shows that there is no significant difference between the solutions obtained by applying difference α_{Fl}. Moreover, with the same α_{Fl}, the object value increase while the α_{Cu} decrease.

Now, we compare the results obtained by using different fitness functions of instances. Although there is no significant difference between the result obtained by using Cu grade chance constraint and combine chance constraints separately for the same instance with the same level of α_{Cu}. By comparing the solutions list in the column *Fl Chance constraint* in Table 3 against that of the combined chance constraint in the same value of α_{Fl}, we find that for the same instance, in most case the results obtained by a single chance constraint are better than the multi chance constraints.

One interesting finding is that the value of α_{Fl} does not show significant effects on the results in the experiments. A possible explanation for this might be that the parameters of the instances are not reliable or match the real-world situation, which can indicate the malfunction of the constraint. This is an important issue for feature research. To develop approaches used to creates a benchmark that more reliable or more close to the real-world situation for the stockpile blending problem.

6 CONCLUSION

Chance-constrained optimization problems play a crucial role in various real-world applications as they allow to limit the probability of violating a given constraint when dealing with stochastic problems. In this paper, we consider the stockpile blending problem which is an important component in mine scheduling with the uncertainty in the geologic input data. We modeled the stockpile blending problem as a nonlinear optimization problem and introduced the chance constraints to tackle the uncertainty with the assumption that the material grades are stochastic variables. We show how to incorporate a well-known probability tail, the Chebyshev's inequality, into presenting the surrogate functions of the chance constraints. Furthermore, we designed the four fitness functions with different chance constraints. In our experiments, which have covered a variety of instances according to the parameters, we have observed that the probability of the Cu chance constraint affects the results obtained by using the fitness function only consider the Cu chance constraint and the fitness function with considering multi chance constraints. Due to the ineffectiveness of the probability of the Fl chance constraint, for further studies, it could be interesting to deeply investigate the relationship between chance constraints. It would be also interesting to develop benchmarks for the stockpile blending problem with chance constraints as there is no available open access data-set.

ACKNOWLEDGEMENTS

This research has been supported by the SA Government through the PRIF RCP Industry Consortium.

REFERENCES

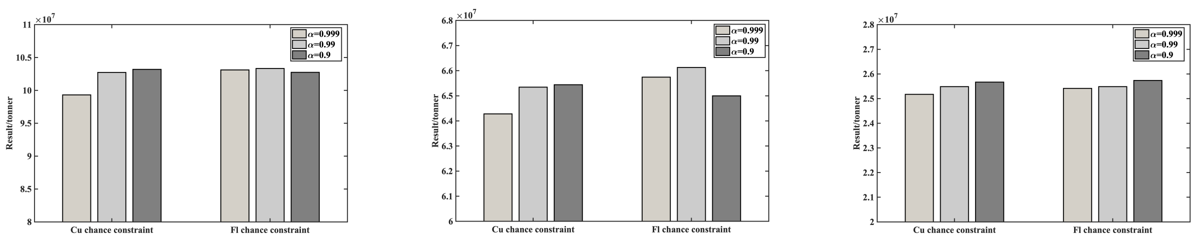
[1] Atsushi Akaike and Kadri Dagdelen. 1999. A strategic production scheduling method for an open pit mine. *proceedings of the 28th Application of Computers and Operation Research in the Mineral Industry* (1999), 729–738.
[2] Mohammad Waqar Ali Asad and Roussos Dimitrakopoulos. 2012. Optimal production scale of open pit mining operations with uncertain metal supply and long-term stockpiles. *Resources policy* 37, 1 (2012), 81–89.
[3] Hirad Assimi, Oscar Harper, Yue Xie, Aneta Neumann, and Frank Neumann. 2020. Evolutionary Bi-Objective Optimization for the Dynamic Chance-Constrained Knapsack Problem Based on Tail Bound Objectives. In *ECAI 2020 - 24th European Conference on Artificial Intelligence*. IOS Press, 307–314.
[4] CK Baker and SM Giacomo. 1998. Resource and reserves: their uses and abuses by the equity markets. In *Ore reserves and finance: a joint seminar between Australasian Institute of Mining and Metallurgy (AusIMM) and Australian Securities Exchange (ASX), Sydney*.
[5] Andreas Bley, Natashia Boland, Gary Froyland, and Mark Zuckerberg. 2012. Solving mixed integer nonlinear programming problems for mine production planning with stockpiling. *Preprint at http://www.optimization-online.org/DB_HTML/2012/11/3674.html)* (2012).
[6] Abraham Charnes and William W Cooper. 1959. Chance-constrained programming. *Management science* 6, 1 (1959), 73–79.
[7] Swagatam Das, Sankha Subhra Mullick, and Ponnuthurai N Suganthan. 2016. Recent advances in differential evolution–an updated survey. *Swarm and Evolutionary Computation* 27 (2016), 1–30.
[8] Swagatam Das and Ponnuthurai Nagaratnam Suganthan. 2010. Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation* 15, 1 (2010), 4–31.
[9] Benjamin Doerr, Carola Doerr, Aneta Neumann, Frank Neumann, and Andrew M. Sutton. 2020. Optimization of Chance-Constrained Submodular Functions. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*. AAAI Press, 1460–1467.

Table 3: Fitness values obtained with single chance constraint

Instance		Deterministic	Cu Chance constraint (α_{Cu})			Fl Chance constraint (α_{Fl})		
			0.999	0.99	0.9	0.999	0.99	0.9
1	Mean	103603035.94	99319128.52	102724900.09	103206748.35	103117715.98	103340755.20	102753876.59
	Best	110830487.20	100434268.90	110489368.20	111221777.51	108460913.10	110593860.12	106976825.01
	Worst	100025173.10	98404158.16	99426947.62	99935646.96	99979453.63	98951340.30	99444063.80
	Success rate		0.166666667	1	1	1	1	1
2	Mean	66339866.34	64280794.53	65346088.50	65440690.91	65741114.11	66128957.43	64999603.10
	Best	69691652.87	66062865.43	70504938.14	69307609.02	70846603.30	69265095.80	67416065.20
	Worst	63401302.85	62822049.78	61409848.38	62043167.43	62885220.60	63139531.31	62369471.32
	Success rate		0.3	1	1	1	1	1
3	Mean	25706739.82	25172345.85	25484058.55	25667396.95	25414602.50	25487090.60	25737554.80
	Best	26714591.61	25780112.93	27501228.19	26652385.99	26542657.00	26440088.30	27420748.21
	Worst	25042412.92	24939884.30	24517912.31	24338065.15	24301347.20	24502516.90	24675079.20
	Success rate		0.166666667	0.733333333	0.8	0.83333333	0.7	1

Table 4: Fitness values obtained with two chance constraints

Instance		Combine Chance constraints								
		α_{Cu} = 0.999			α_{Cu} = 0.99			α_{Cu} = 0.9		
α_{Fl}		0.999	0.99	0.9	0.999	0.99	0.9	0.999	0.99	0.9
1	Mean	98787701.39	99241024.07	99492631.19	102600300.50	102276579.70	102682519.61	103388918.00	102934747.10	102493177.23
	Best	102031085.80	103755489.91	102599092.62	106167319.20	107035409.13	109936137.61	106369047.00	108285976.13	108115669.08
	Worst	96585053.09	97115231.16	97595258.99	99322445.27	98370524.81	100035474.00	100191353.05	99131474.70	99124405.82
	Success rate	0.366666667	0.366666667	0.4	1	1	1	1	1	1
2	Mean	65983255.56	63563990.36	64902632.44	65902068.31	65729892.86	65506695.73	65678975.14	65559860.50	65651773.33
	Best	69079010.41	65782360.39	69006540.86	69991791.36	69754992.24	70596782.19	69703718.21	68258615.00	70556638.81
	Worst	62870362.84	61877215.08	61963922.28	59672821.89	63038071.83	61823714.56	61900565.70	61618562.90	62959270.15
	Success rate	0.166666667	0.133333333	0.3	1	1	1	1	1	1
3	Mean	25960917.39	25871636.05	25686417.52	25459522.87	25682180.87	25687541.01	25569238.21	25498602.00	25617280.63
	Best	25960917.39	26273134.25	25686417.52	26575050.89	26643851.74	26329733.68	26457335.14	26589310.80	26326709.05
	Worst	25960917.39	25473933.01	25686417.52	24495873.72	24826217.28	24668501.30	24693215.90	24721583.71	24549825.10
	Success rate	0.033333333	0.133333333	0.033333333	0.8	0.833333333	0.866666667	0.8	0.86666667	0.76666667



(a) Instance 1 (b) Instance 2 (c) Instance 3

Figure 1: Bar graph for DE algorithm with single chance constraint

[10] Thys B Johnson. 1968. *Optimum open pit mine production scheduling*. Technical Report. California Univ Berkeley Operations Research Center.

[11] K Jupp, TJ Howard, and JE Everett. 2013. Role of pre-crusher stockpiling for grade control in iron ore mining. *Applied Earth Science* 122, 4 (2013), 242–255.

[12] Nir Lipovetzky, Christina N. Burt, Adrian R. Pearce, and Peter J. Stuckey. 2014. Planning for Mining Operations with Time and Resource Constraints. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling, ICAPS 2014*.

[13] Baoding Liu. 2007. Uncertainty theory. In *Uncertainty theory*. Springer, 205–234.

[14] Bo Liu, Qingfu Zhang, Francisco V. Fernández, and Georges G. E. Gielen. 2013. An efficient evolutionary algorithm for chance-constrained bi-objective stochastic optimization. *IEEE Trans. Evolutionary Computation* 17, 6 (2013), 786–796.

[15] T. McConaghy, P. Palmers, M. Steyaert, and G. G. E. Gielen. 2009. Variation-aware structural synthesis of analog circuits via hierarchical building blocks and structural homotopy. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 28, 9 (2009), 1281–1294.

[16] Lei L Mercado, S-M Kuo, Tien-Yu Lee, and Russell Lee. 2005. Analysis of RF MEMS switch packaging process for yield improvement. *IEEE transactions on advanced packaging* 28, 1 (2005), 134–141.

[17] Bruce L Miller and Harvey M Wagner. 1965. Chance constrained programming with joint constraints. *Operations Research* 13, 6 (1965), 930–945.

[18] Eduardo Moreno, Mojtaba Rezakhah, Alexandra Newman, and Felipe Ferreira. 2017. Linear models for stockpiling in open-pit mine production scheduling problems. *European Journal of Operational Research* 260, 1 (2017), 212–221.

[19] Ferrante Neri and Ville Tirronen. 2010. Recent advances in differential evolution: a survey and experimental analysis. *Artificial Intelligence Review* 33, 1-2 (2010), 61–106.

[20] Aneta Neumann and Frank Neumann. 2020. Optimising Monotone Chance-Constrained Submodular Functions Using Evolutionary Multi-objective Algorithms. In *Parallel Problem Solving from Nature - PPSN XVI - 16th International Conference, PPSN 2020, Proceedings, Part I*. 404–417.

[21] Karol R Opara and Jarosław Arabas. 2019. Differential Evolution: A survey of theoretical analyses. *Swarm and evolutionary computation* 44 (2019), 546–558.

[22] M Osanloo, J Gholamnejad, and B Karimi. 2008. Long-term open pit mine production planning: a review of models and algorithms. *International Journal of Mining, Reclamation and Environment* 22, 1 (2008), 3–35.

[23] Nam Pham, A Malinowski, and T Bartczak. 2011. Comparative study of derivative free optimization algorithms. *IEEE Transactions on Industrial Informatics* 7, 4 (2011), 592–600.

[24] Chandra A Poojari and Boby Varghese. 2008. Genetic algorithm based technique for solving chance constrained problems. *European journal of operational research* 185, 3 (2008), 1128–1154.

[25] Kenneth V Price. 1996. Differential evolution: a fast and simple numerical optimizer. In *Proceedings of North American Fuzzy Information Processing*. IEEE, 524–527.

[26] Mojtaba Rezakhah and Eduardo Moreno. 2019. Open Pit Mine Scheduling Model Considering Blending and Stockpiling. In *International Symposium on Mine Planning & Equipment Selection*. Springer, 75–82.

[27] Mojtaba Rezakhah, Eduardo Moreno, and Alexandra Newman. 2020. Practical performance of an open pit mine scheduling model considering blending and stockpiling. *Computers & Operations Research* 115 (2020), 104638.

[28] GK Robinson. 2004. How much would a blending stockpile reduce variation? *Chemometrics and intelligent laboratory systems* 74, 1 (2004), 121–133.

[29] Mehran Samavati, Daryl Essam, Micah Nehring, and Ruhul Sarker. 2017. A local branching heuristic for the open pit mine production scheduling problem. *European Journal of Operational Research* 257, 1 (2017), 261–271.

[30] Farzad Sotoudeh, Micah Nehring, Mehmet Kizil, Peter Knights, and Amin Mousavi. 2020. Production scheduling optimisation for sublevel stoping mines using mathematical programming: A review of literature and future directions. *Resources Policy* 68 (2020), 101809.

[31] Rainer Storn. 1996. On the usage of differential evolution for function optimization. In *Proceedings of North American Fuzzy Information Processing*. IEEE, 519–523.

[32] Rainer Storn and Kenneth Price. 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* 11, 4 (1997), 341–359.

[33] Mohammad Tabesh, Hooman Askari-Nasab, and Rodrigo Peroni. 2015. A comprehensive approach to strategic open pit mine planning with stockpile consideration. In *Proceedings of the Thirty-seventh international symposium on applications of computers and operations research in mineral industry*. Society for Mining, Metallurgy and Exploration, 326–332.

[34] E. Topal and S. Ramazan. 2012. Strategic mine planning model using network flow model and real case application. *International Journal of Mining, Reclamation and Environment* 26, 1 (2012), 29–37.

[35] Stanislav Uryasev. 2013. *Probabilistic constrained optimization: methodology and applications*. Vol. 49. Springer Science & Business Media.

[36] Yue Xie, Oscar Harper, Hirad Assimi, Aneta Neumann, and Frank Neumann. 2019. Evolutionary algorithms for the chance-constrained knapsack problem. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019*. ACM, 338–346.

[37] Yue Xie, Aneta Neumann, and Frank Neumann. 2020. Specific single- and multi-objective evolutionary algorithms for the chance-constrained knapsack problem. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2020*. ACM, 271–279.