

EDYTA BRZYCHCZY*

Tradycyjne metody optymalizacji a nowoczesna heurystyka w wybranych zagadnieniach modelowania robót górniczych w kopalniach węgla kamiennego

Wprowadzenie

Przedmiotem podjętych badań jest zagadnienie planowania robót eksploatacyjnych w wielozakładowym przedsiębiorstwie górniczym. Należy zauważyć, że w warunkach obecnie funkcjonujących przedsiębiorstw, zakres tego planowania dotyczy możliwości wyposażenia planowanych robót, ponieważ parametry geometryczne pól ścianowych jak i kolejność ich wybierania są znacznie wyprzedzająco zapisywane i akceptowane w sporządzanych planach techniczno-ekonomicznych.

Zagadnienie wyposażenia przodków ścianowych jest bardzo istotne z uwagi na:

- kosztowność tego typu maszyn i urządzeń,
- duży potencjał środków produkcyjnych będących w gestii przedsiębiorstwa przy jednoczesnym zróżnicowaniu warunków górniczo-geologicznych poszczególnych zakładów (kopalni),
- częste przypadki ograniczania możliwości produkcyjnych maszyn i urządzeń z powodu trudnych warunków pracy,
- możliwość zwiększenia efektywności posiadanych środków produkcyjnych w warunkach wielozakładowego przedsiębiorstwa górniczego poprzez odpowiednie rozmieszczenie wyposażenia pomiędzy zakładami (kopalniami), co wykazano między innymi w pracy (Klank 2002).

Przedmiotem prac autorki stało się poszukiwanie metody, która wspomże planowanie technicznego wyposażenia przodków ścianowych w kopalniach wchodzących w skład większego podmiotu górniczego (spółki) z uwzględnieniem kryterium efektywności wykorzystania – posiadanych i potencjalnie dostępnych – maszyn i urządzeń.

1. Ogólna charakterystyka analizowanego zagadnienia

Problem, który został powyżej przedstawiony można określić jako przypadek znany z kombinatoryki pod nazwą wariacja z powtórzeniami. Wariacją (rozmięszczeniem) bowiem z powtórzeniami z n elementów po k nazywamy uporządkowany zbiór składający się z k elementów różnych lub nie różniących się między sobą, wybranych spośród n różnych elementów. Tak więc dla k przodków ścianowych wybieramy wyposażenie z n zestawów ścianowych.

Przyjmując, że:

- 1) przedsiębiorstwo składa się z m kopalni,
 - 2) w analizowanym okresie w każdej kopalni jest l_i przodków ścianowych przewidzianych do eksploatacji,
 - 3) przedsiębiorstwo deklaruje posiadanie bądź pozyskanie z zestawów ścianowych,
 - 4) liczba zestawów, które mogą zostać wykorzystane w danym przodku ze względu na warunki górniczo-geologiczne jest ograniczona do połowy, czyli $n = z/2$
- to liczbę możliwych wariantów wyposażenia możemy określić ze wzoru (Gerstenkorn, Śródka 1976):

$$V_n^k = n^k \quad (1)$$

gdzie:

$$k = \sum_{i=1}^m l_i$$

Aby móc zrozumieć złożoność i objętość obliczeniową przedstawionego zagadnienia można rozpatryć następujący przykład:

Przedsiębiorstwo X składa się z 4 kopalń, w każdej z nich planowane jest wyposażenie 3 przodków ścianowych (prowadzenie których rozłożone jest w czasie), a liczba posiadanych i dostępnych zestawów ścianowych wynosi 14 (ich dostępność jest zmienna w czasie). Jeżeli zostanie przyjęte założenie, że tylko połowa w danym momencie jest dostępna dla danego przodka, zarówno ze względu na moment rozpoczęcia jego prowadzenia, jak i z uwagi na warunki górniczo-geologiczne, to całkowita liczba wariantów, którą należałoby przeanalizować wynosi $V_7^{12} = 7^{12} = 13\ 841\ 287\ 200$. Wprowadzając ograniczenie dopuszczalności wariantów ze względu na istotne założenia związane z całkowitym czasem

trwania planowanych robót – czasem niezbędnym na przebrojenie i likwidację, które może spowodować zmniejszenie ich liczby nawet o 90% – to nadal do analizy pozostaje ponad 1,3 mln wariantów. Stąd przedmiotem poszukiwań autorki była metoda optymalizacji, która w efektywny sposób umożliwi relatywnie szybkie przeszukiwanie przestrzeni rozwiązań dla sformułowanego problemu.

W tym miejscu należałoby podkreślić kilka kwestii, które w dość znacznym stopniu wpływają na metodę rozwiązania postawionego problemu (Michalewicz, Fogel 2006):

- 1) złożone problemy często mają olbrzymią liczbę możliwych rozwiązań,
- 2) żeby dojść do rozwiązania musimy często posługiwać się uproszczonym modelem, który może uniemożliwić rozwiązanie właściwego problemu,
- 3) warunki problemu zmieniają się w czasie,
- 4) rzeczywiste problemy często mają ograniczenia, które wymagają specjalnych operacji do generowania rozwiązań dopuszczalnych.

Istotnymi elementami każdego algorytmu, wymagającymi podkreślenia są: odpowiednio sformułowany model, jego postać i przyjęte przy jego sformułowaniu założenia. Każdy model stanowi pewne uproszczenie rzeczywistego problemu, niemniej jednak dla każdej algorytmicznej metody rozwiązywania problemów ważne jest, aby właściwie określał:

- reprezentację problemu,
- cel modelu,
- funkcję oceny rozwiązań.

Wymienione powyżej elementy wpływają znacząco na dobór metody rozwiązania, odziorowanego za pomocą modelu problemu.

1.1. Model problemu

Dla przedstawionego przykładu określono następujące właściwości sformułowanego modelu:

- reprezentacja problemu – ciąg liczb rzeczywistych (o długości k) w kolejności odpowiadającej analizowanym przodkom ścianowym. Liczby rzeczywiste to wartość postępu n -tego zestawu ścianowego w k -tym przodku ścianowym (pos [m/d]),
- celem modelu jest otrzymanie optymalnego rozłożenia n zestawów w k przodkach ścianowych,
- funkcja oceny rozwiązań jest dość złożona, a jej fragment można zapisać w postaci:

$$f_{oceny} = \begin{cases} Wden_{sr} \rightarrow Wden_{plan} \\ kj_{sr} \rightarrow \min \end{cases} \quad (2)$$

gdzie:

- $Wden_{sr}$ – wartość średnia wydobywania wszystkich przodków ścianowych eksploatowanych w czasie t ,

- $Wden_{plan}$ – wartość wydobywania planowana w czasie t ,
 kj_{sr} – wartość średnia kosztu jednostkowego sprzedanego węgla w analizowanym okresie.

Z uwagi na objętość wzorów dla wyprowadzenia powyższych postaci funkcji w tym miejscu tylko krótko zostaną przedstawione najważniejsze zależności pomiędzy elementami składowymi funkcji a wartością postępu zawartą w reprezentacji problemu oraz, w pewnym uproszczeniu, tok generowania rozwiązania:

- w etapie pierwszym następuje losowanie k zestawów ze zbioru n zestawów do określonych przodków ścianowych,
- wybrane przyporządkowania porównuje się z bazą danych o postępach zestawów ścianowych w przodkach prowadzonych w przeszłości (wykorzystuje się w tym celu wielowymiarową analizę porównawczą),
- z bazy danych dla podobnych przodków i zestawów w nich pracujących przyjmuje się funkcję rozkładu prawdopodobieństwa postępu robót eksploatacyjnych oraz funkcję kosztu z nim związaną, o czym pisano m.in. w (Brzychczy 2007),
- wartość postępu (pos) generowana jest podczas obliczeń symulacyjnych z przyjętego rozkładu prawdopodobieństwa,
- w każdym z przodków ścianowych dla wylosowanej wartości postępu oblicza się:

$$t_e = \frac{w}{pos} \quad [\text{d}] \quad (3)$$

$$Wden_d = pos \cdot ls \cdot h \cdot \gamma \cdot k_{ep} \quad [\text{t/d}] \quad (4)$$

$$koszt_d = A \cdot pos + B \quad [\text{zł/d}] \quad (5)$$

gdzie:

- t_e – czas trwania robót eksploatacyjnych w danym przodku [d],
- w – wybieg ściany [m],
- $Wden_d$ – wydobywanie dzienne netto z robót eksploatacyjnych w danym przodku,
- ls – długość ściany [m],
- h – wysokość ściany [m],
- γ – ciężar objętościowy węgla [t/m^3],
- k_{ep} – współczynnik strat eksploatacyjnych i przerobczych,
- $koszt_d$ – dzienny koszt prowadzenia robót eksploatacyjnych w danym przodku [zł/d],
- A – współczynnik obejmujący koszty zależne od postępu przodka [zł/m],
- B – współczynnik obejmujący koszty niezależne od postępu przodka [zł/d],

- czas trwania robót eksploatacyjnych przenoszony jest na sieć czynności, która odwzorowuje rozłożenie poszczególnych przodków w czasie (dodatkowo obliczony czas zostaje przeliczony na dni kalendarzowe),
- sieć czynności zostaje uzupełniona o czas trwania robót zbrojeniowych i likwidacyjnych,
- dla wybranego czasu t (np. miesiąc, kwartał) oblicza się za pomocą symulacji $Wden_{sr}$, które powinny być zbliżone do wartości planowanych w tych elementarnych odcinkach czasu,
- natomiast aby wyznaczyć kj_{sr} w pierwszej kolejności należy zsumować koszty robót eksploatacyjnych (oraz robót zbrojeniowo-likwidacyjnych) w danym dniu roboczym, a ich wielkość odnieść się do wydobycia dobowego z prowadzonych przodków, a po wykonaniu odpowiedniej liczby iteracji określić wartość średnią dla analizowanego okresu.

Powyższe rozważania w bardzo uproszczony i wybiórczy sposób przedstawiają model, dla którego poszukiwana jest metoda rozwiązania sformułowanego problemu, a z uwagi na to, że funkcja oceny jest wielokryterialna (co tylko fragmentarycznie przedstawiono wzorem 2), poszukiwania takie nie będą łatwe.

2. Metody klasyczne

Istnieje wiele klasycznych algorytmów poszukiwania optymalnego rozwiązania w danej przestrzeni rozwiązań, jednak żadna z tradycyjnych metod nie jest wystarczająco elastyczna (za każdym razem, gdy zmienia się problem, musi zostać zmieniony model). Jednakże „*klasyczna metoda optymalizacyjna, jeśli odpowiada zadaniu, które rozwiązujemy, może dać bardzo dobre wyniki*” – takie stwierdzenie pada w pracy (Michalewicz, Fogel 2006), w której zamieszczono również podział takich metod na dwie klasy:

- 1) algorytmy, które oceniają tylko pełne rozwiązania,
- 2) algorytmy, które wymagają oceny rozwiązań częściowo skonstruowanych (niepełne rozwiązania pierwotnego problemu) lub przybliżonych (pełne rozwiązanie zredukowanego problemu).

Do algorytmów oceniających pełne rozwiązanie można zaliczyć między innymi: przeszukiwanie wyczerpujące, przeszukiwanie lokalne oraz metody z zakresu programowania matematycznego (np. programowanie liniowe, nieliniowe). Zaletą działania na pełnych rozwiązaniach jest fakt, że w przypadku przerwania algorytmu otrzymuje się jakiegokolwiek, potencjalne rozwiązanie sformułowanego problemu. Natomiast pewne wady ich stosowania wynikają z konstrukcji wymienionych procedur.

Krótkie zestawienie cech wybranych algorytmów, ze względu na możliwość ich zastosowania do sformułowanego problemu, przedstawiono w tabeli 1.

Do rozwiązania problemu przedstawionego w przykładzie – ze względu na czasochłonność procesu obliczeniowego – na pewno nie zostanie wybrana metoda przeszukiwania

Charakterystyka wybranych klasycznych metod optymalizacji

TABLE 1

Characteristic of some classic optimization methods

| Algorytm | Krótki opis | Zalety/Wymagania | Wady/Wymagania |
|---|---|--|---|
| Przeszukiwanie wyczerpujące | sprawdzenie każdego rozwiązania z przestrzeni rozwiązań | – wybór globalnie najlepszego rozwiązania, nie jest ważna kolejność generowania rozwiązań | – całkowita przestrzeń rozwiązań jest najczęściej olbrzymia, – znaczna czasochłonność poszukiwań (w zależności od złożoności problemu kilka lub nawet kilkadziesiąt lat pracy komputera) |
| Przeszukiwanie lokalne | przeszukiwanie części przestrzeni rozwiązań w otoczeniu konkretnego rozwiązania | – mniejsza czasochłonność poszukiwań, jeśli rozmiar otoczenia jest niewielki to możemy je bardzo szybko przeszukać | – istotny wpływ na kierunek poszukiwania ma postać przekształcenia bieżącego rozwiązania, – „pułapka” lokalnego optimum |
| Programowanie liniowe (metoda sympleks) | znalezienie ekstremum liniowej kombinacji zmiennych przy odpowiednich ograniczeniach oraz określonych warunkach | – poszukiwanie prowadzi się w kolejnych etapach, na pełnych i cząstkowych rozwiązaniach | – funkcja oceny powinna być wyrażona w postaci liniowej kombinacji zmiennych – jeżeli problem jest złożony, pojawia się duża liczba zmiennych i równań, co wydłuża czas rozwiązania problemu |
| Programowanie nieliniowe | znalezienie ekstremum kombinacji zmiennych jeżeli funkcja celu lub chociaż jeden z warunków ograniczających jest funkcją nieliniową | – poszukiwanie prowadzi się w kolejnych etapach, na pełnych i cząstkowych rozwiązaniach | – brak ogólnej metody rozwiązywania zadań nieliniowych – często brak jest odpowiednich danych aby ustalić postać analityczną badanych zależności |

Źródło: opracowanie własne na podstawie (Michalewicz, Fogel 2006; Badania... 1996; Grabowski 1982)

wyczerpującego. Jeżeli bowiem przyjmie się założenie, że wygenerowanie jednego rozwiązania zajmuje 1 sekundę (co jest raczej zaniżoną wartością biorąc pod uwagę złożoną postać funkcji oceny), to wywołanie pełnego zestawu rozwiązań z przykładu zajęłoby 445 miesięcy (!), a przy ograniczeniu do 10% rozwiązań dopuszczalnych tylko 15 dni (!). Można przypuszczać, że dla większości projektantów praca takiego algorytmu nie byłaby zachęcająca, nie wspominając o potrzebach np. powtórnych symulacji.

Metoda przeszukiwania lokalnego jest bardzo obiecująca, jednak przy tak olbrzymiej przestrzeni rozwiązań będzie ona również czasochłonna (czas liczony nie w dniach, ale

raczej w godzinach), a przeszukiwania przestrzeni rozpoczęte od konkretnego (jednego) rozwiązania mogą utknąć w lokalnym optimum.

Wielokryterialna funkcja oceny wyklucza wykorzystanie metod programowania liniowego do poszukiwania najlepszego rozwiązania; ten element wpływa również na nieprzydatność pozostałych dwóch metod w przedstawionych powyżej pierwotnych postaciach. Postać i przyjęte ograniczenia dla sformułowanego modelu wykluczają również zastosowanie programowania nieliniowego.

TABELA 2

Charakterystyka klasycznych metod optymalizacji dla rozwiązań częściowych

TABLE 2

Characteristic of classic optimization methods for partial solutions

| Algorytm | Krótki opis | Zalety/Wymagania | Wady/Wymagania |
|------------------------------|--|---|--|
| Algorytmy zachłanne | algorytm tworzy pełne rozwiązanie za pomocą ciągu kroków | – prostota – należy przypisywać wartości wszystkim zmiennym problemu podejmując za każdym razem możliwie najlepszą decyzję | – podejmowanie najlepszej decyzji w każdym poszczególnym kroku nie zawsze musi prowadzić do globalnego optymalnego rozwiązania |
| Dziel i rządź | algorytm polega na podziale problemu na mniejsze części i ich rozwiązaniu | – podejście jest efektywne jeśli czas i wysiłek potrzebny na wykonanie podziału, obliczeń i złożenie odpowiedzi jest mniejszy niż rozwiązanie pierwotnej postaci modelu | – nie zawsze jest możliwe złożenie pełnego rozwiązanie po dokonaniu podziału na mniejsze części |
| Programowanie dynamiczne | algorytm znajduje pełne rozwiązanie na podstawie pośredniego punktu, pomiędzy punktem bieżącym a celem | – algorytm znajduje rozwiązanie problemów, dla których zawadzą algorytmy zachłanne | – często duża złożoność obliczeniowa, – algorytmy niekiedy trudne do zrozumienia, bowiem ich konstrukcja zależy od specyfiki problemu |
| Metoda podziału i ograniczeń | algorytm korzysta z określonej koncepcji dzielenia przestrzeni przeszukiwania | – szybkość algorytmu uzyskuje się dzięki odpowiednio wprowadzonym ograniczeniom | – konieczność określenia dolnego lub górnego ograniczenia poszczególnych rozwiązań – przegląd przestrzeni rozwiązań należy odpowiednio ukierunkować |
| Algorytm A* | algorytm działa na grafach o różnej strukturze, główna koncepcja – najpierw najlepszy | – algorytm korzysta z dwóch list wierzchołków otwartych i zamkniętych, przeszukiwanie bada najbardziej obiecujący wierzchołek – poprawnie skonstruowany algorytm gwarantuje uzyskanie globalnego rozwiązania | – przeszukiwanie typu „najpierw najlepszy” korzysta z heurystyki określającej wartość wierzchołka, konieczne jest zatem jej prawidłowe opracowanie |

Przyjęta reprezentacja przedstawionego w przykładzie problemu intuicyjnie kieruje zainteresowanie na algorytmy działające na rozwiązaniach pełnych, lecz warto również przejrzeć wybrane metody klasyczne, które działają na rozwiązaniach niepełnych.

Zestawienie wybranych metod z tego zakresu przedstawiono w tabeli 2.

Metody działające na rozwiązaniach niepełnych we wąskim zakresie umożliwiają szybkie i wiarygodne uzyskanie globalnego rozwiązania. Z drugiej strony metody działające na pełnych rozwiązaniach gwarantują osiągnięcie sukcesu w poszukiwaniach, lecz niestety jest on często okupiony zbyt dużą czasochłonnością. Stąd też wysiłki projektantów skierowane są na opracowanie algorytmów unikających pułapki lokalnych optimum. Przykładami takich metod są między innymi: symulowane wyżarzanie oraz poszukiwanie z tabu. Ich krótką charakterystykę przedstawiono w tabeli 3.

TABELA 3

Algorytmy unikające lokalnych optimum

TABLE 3

Algorithms avoiding local optima

| Algorytm | Krótki opis |
|-----------------------|---|
| Symulowane wyżarzanie | w algorytmie podobnym do przeszukiwania lokalnego wprowadza się dodatkowy parametr zwany temperaturą, który zmienia prawdopodobieństwo przejścia z jednego punktu przeszukiwanej przestrzeni do innego; algorytm kończy przeszukiwanie po wypełnieniu warunku zakończenia, a nie znalezienia ulepszenia ostatniego punktu; przeszukiwanie stochastyczne |
| Poszukiwanie z tabu | algorytm o podobnej strukturze jak symulowane wyżarzanie, niemniej jednak procedura ulepszenia korzysta z historii przeszukiwania, w której ostatnio odwiedzane punkty są punktami tabu i są pomijane podczas podejmowania decyzji o wyborze kolejnego punktu; przeszukiwanie zdeterminowane |

Źródło: opracowanie własne na podstawie (Michalewicz, Fogel 2006)

Algorytmy przedstawione w tabeli 3 działają na pełnych rozwiązaniach, niemniej jednak w odróżnieniu od lokalnego przeszukiwania mają więcej parametrów, które trzeba kontrolować, tj. temperatura, pamięć.

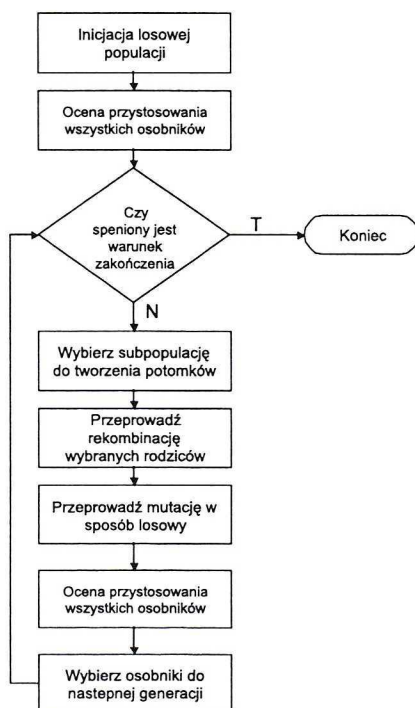
Omówione powyżej metody zarówno działające na rozwiązaniach pełnych oraz częściowych przeszukują przestrzeń rozwiązań od określonego punktu (rozwiązania). Aby zwiększyć efektywność przeszukiwania przestrzeni rozwiązań można rozpocząć przeszukiwanie z wielu punktów (rozwiązań) stanowiących populację rozwiązań, czyli wykorzystać tzw. podejście ewolucyjne.

3. Podejście ewolucyjne

Algorytmy ewolucyjne naśladujące proces ewolucji zachodzący wśród organizmów żywych są bardzo często wykorzystywane do analizy i poszukiwań najlepszego rozwiązania złożonych zagadnień z różnych dziedzin nauki i techniki.

Do podstawowych pojęć używanych w EA należą:

- osobnik czyli propozycja rozwiązania postawionego problemu,
- populacja czyli zbiór osobników o określonej liczebności,
- przystosowanie osobnika czyli wartość liczbową, określająca jakość reprezentowanego przez niego przystosowania (Arabas 2004),
- genotyp określany jako struktura osobnika (Goldberg 2003) bądź punkt w przestrzeni kodów (Arabas 2004),
- fenotyp określany jako zbiór parametrów, rozwiązanie (Goldberg 2003) lub jako zestaw określonych cech poddawanych ocenie środowiska (Arabas 2004),
- chromosom określany jako ciąg kodowy (Goldberg 2003),
- gen czyli składowy element chromosomu (cecha),
- allel czyli możliwa wartość lub wariant cechy (Goldberg 2003).



Rys. 1. Algorytm ewolucyjny – przykład
Źródło: Pawlak 1999

Fig. 1. Evolutionary algorithm – example

Na podstawowy algorytm ewolucyjny zazwyczaj składają się następujące etapy:

- wybór reprezentacji problemu (sposób kodowania),
- wybór sposobu inicjalizacji populacji,
- zastosowanie operatorów genetycznych:
 - krzyżowanie (rekombinacja),
 - mutacja;
- ocena przystosowania osobników,
- selekcja osobników.

Skonstruowany algorytm ewolucyjny może działać na rozwiązaniach pełnych lub częściowych. Uproszczony schemat działania algorytmu ewolucyjnego przedstawiono na rysunku 1.

Ideą skonstruowanego algorytmu jest poszukiwanie najlepszego rozwiązania (osobnika) z analizowanej populacji osobników. Osobniki te rozmnażają się, tworząc nowe osobniki z wykorzystaniem odpowiednich operatorów genetycznych. Do kolejnych etapów ewolucji największe prawdopodobieństwo przejścia mają osobniki najlepiej przystosowane, co określa wartość przyjętej funkcji przystosowania.

Istnieje kilka typów algorytmów ewolucyjnych, których charakterystyczne cechy wraz z postępowaniem tej dziedziny nauki ulegają niekiedy zatarciu. Najczęściej jednak można spotkać się z podziałem algorytmów ewolucyjnych na (Pawlak 1999):

- algorytmy genetyczne,
- programowanie ewolucyjne,
- strategie ewolucyjne,
- programowanie genetyczne.

3.1. Algorytmy genetyczne

W algorytmach genetycznych populacja osobników reprezentowana jest przez łańcuchy znaków binarnych (0 i 1). W praktyce można wykorzystać ten rodzaj algorytmów jeżeli potrafimy utworzyć łańcuchy binarne reprezentujące rozwiązania problemu. Chromosom z kodowaniem binarnym jest więc wektorem n -genów o wartościach 0 lub 1. Schemat prostego algorytmu genetycznego przedstawiono na rysunku 2.

Algorytm analizuje dwie populacje: P^t zwaną populacją bazową oraz O^t zwaną populacją potomną. Wprowadzona jest również populacja tymczasowa T^t . W wymienionych populacjach liczba osobników jest taka sama.

W chwili inicjacji populacja bazowa jest wypełniana losowo wygenerowanymi osobnikami (P^0). Dla każdego z osobników jest obliczana wartość funkcji przystosowania. W kolejnym etapie rozpoczyna się proces ewolucji.

W pierwszej kolejności następuje reprodukcja (selekcja), która polega na skopiowaniu losowo wybranych osobników do populacji tymczasowej (każdego osobnika charakteryzuje inne prawdopodobieństwo wylosowania zależne od wartości funkcji przystosowania). Populacja tymczasowa najczęściej tworzona jest na podstawie selekcji ruletkowej (inaczej zwanej proporcjonalną).

```

procedure Prosty algorytm genetyczny
begin
  t:= 0
  inicjacja  $P^0$ 
  ocena  $P^0$ 
  while (not warunek stopu) do
    begin
       $T^t$ := reprodukcja  $P^t$ 
       $O^t$ :=krzyżowanie i mutacja  $T^t$ 
      ocena  $O^t$ 
       $P^{t+1}$ :=  $O^t$ 
      t:= t+1
    end
  end

```

Rys. 2. Schemat prostego algorytmu genetycznego
Źródło: (Arabas 2004)

Fig. 2. Scheme of simple genetic algorithm

W kolejnym etapie osobniki są poddawane krzyżowaniu i mutacji. Prawdopodobieństwo krzyżowania p_c jest jednym z parametrów algorytmu. Chromosomy rodzicielskie są rozcinane na dwa fragmenty w tym samym miejscu (wybrany losowo z rozkładem równomiernym) i są ze sobą wymiennie łączone (tzn. część pierwsza chromosomu rodzica 1 łączona jest z częścią drugą chromosomu rodzica 2 itd.). Natomiast zadaniem operatora mutacji są losowe zmiany w genotypach osobników (zamiana wartości genu na przeciwną) z prawdopodobieństwem p_m , które również jest jednym z parametrów algorytmu.

Powstałe w ten sposób osobniki poddawane są ocenie i stanowią populację potomną, a ta z kolei staje się populacją bazową w kolejnej pętli obliczeń algorytmu. Obliczenia kontynuuje się do spełnienia warunku zatrzymania algorytmu. Pojedyncza pętla obliczeń zwana jest generacją.

Algorytm genetyczny jest metodą niedeterministyczną i nie ma gwarancji znalezienia za każdym razem identycznego rozwiązania. Wyjaśnienie zasad jego działania opisane jest w literaturze pod nazwą teorii schematów i stanowi stosunkowo nowy rozdział w teorii algorytmów genetycznych.

3.2. Programowanie ewolucyjne

Cechą charakterystyczną programowania ewolucyjnego jest rzeczywistoliczbowa reprezentacja chromosomu. Po inicjalizacji populacji bazowej wszystkie osobniki stają się rodzicami. W jednym kroku algorytmu każdy osobnik tworzy jednego potomka, który jest poddawany mutacji (forma mutacji zależy od przyjętej reprezentacji problemu i jest często adaptacyjna). W ten sposób powstaje populacja potomna. Przykładowy algorytm programowania ewolucyjnego przedstawiono na rysunku 3.

Po ocenie osobników potomnych tworzy się nową populację bazową z wykorzystaniem selekcji rangowej (Arabas 2004) bądź turniejowej (Pawlak 1999), szczegółowo opisanych w podanej literaturze.

```

procedure Programowanie ewolucyjne
begin
  t:= 0
  inicjacja  $P^t$ 
  ocena  $P^t$ 
  while (not warunek stopu) do
    begin
       $O^t := \emptyset$ 
      foreach ( $X \in P^t$ ) do
        begin
          Y := mutacja X
           $O^t := O^t \cup \{Y\}$ 
        end
      end
      ocena  $O^t$ 
      foreach ( $X \in P^t \cup O^t$ ) do
        begin
          wyznacz r(X)
        end
       $P^{t+1} :=$  wybór najwyższych rangą z  $P^t \cup O^t$ 
    end
  end

```

Rys. 3. Schemat programowania ewolucyjnego
Źródło: (Arabas 2004)

Fig. 3. Scheme of evolutionary programming

W programowaniu ewolucyjnym zazwyczaj nie używa się operatora krzyżowania, ponieważ formy mutacji są na tyle elastyczne, że umożliwiają odpowiednie przekształcenie genotypu osobników.

3.3. Strategie ewolucyjne

Pierwszy algorytm strategii ewolucyjnych oparty był na prostym schemacie mutacji i selekcji z wykorzystaniem jednego rodzica, który podlegając mutacjom, tworzył jednego potomka. Jeżeli potomek okazał się lepszy od rodzica, to on stawał się rodzicem. W przeciwnym razie dotychczasowy rodzic pełnił swoją funkcję w kolejnej generacji. Ten najprostszy wariant nosi nazwę strategii (1 + 1).

Podobnie jak ma miejsce w programowaniu ewolucyjnym, reprezentacja chromosomu w strategiach ewolucyjnych jest wektorem liczb rzeczywistych. Algorytm strategii ewolucyjnej (1 + 1) przedstawiono na rysunku 4.

Niestety, strategia ta przejawia niewielką odporność na minima lokalne, co było motywacją do poszukiwania lepszych schematów i tak powstały strategie (1 + λ), (μ + λ), (μ , λ) (Arabas 2004), gdzie λ oznaczono liczbę potomków powstających w wyniku reprodukcji μ rodziców z uwzględnieniem odpowiedniego przekształcenia populacji osobników.

Charakterystyczne dla strategii typu (μ + λ) jest kodowanie osobników i operacje genetyczne jakim są poddawane. Osobnik ma strukturę składającą się z dwóch chromosomów. Pierwszym z nich jest wektor X wartości zmiennych niezależnych, natomiast drugim najczęściej jest wektor σ zawierający wartości standardowych odchyłeń wykorzystywanych podczas mutacji.

```

procedure Strategia ewolucyjna (1+1)
begin
  t:= 0
  inicjacja  $X^t$ 
  ocena  $X^t$ 
  while (not warunek stopu) do
    begin
       $Y^t$ := mutacja  $X^t$ 
      ocena  $Y^t$ 
      if ( $\Phi(Y^t) > \Phi(X^t)$ ) then
        begin
           $X^{t+1}$  :=  $Y^t$ 
        end
      else
           $X^{t+1}$  :=  $X^t$ 
        end
      t:= t+1
    end
  end

```

Rys. 4. Schemat strategii ewolucyjnej (1+1)

Źródło: (Arabas 2004)

Fig. 4. Scheme of evolutionary strategy (1+1)

W strategii $(\mu + \lambda)$ μ rodziców przy użyciu operatora krzyżowania i mutacji produkuje λ potomków, przy czym $\lambda \geq 1$. Spośród całej grupy osobników rodzicielskich i potomnych wybieranych jest μ najlepszych jednostek i tworzą one następną populację bazową.

Natomiast schemat strategii (μ, λ) różni się od omówionego powyżej $(\mu + \lambda)$ tym, że nową populację bazową tworzą tylko najlepsze osobniki potomne, a osobniki rodzicielskie zostają pominięte.

Strategie ewolucyjne wykorzystuje się do problemów optymalizacji wektorowej i mogą one służyć jako heurystyki do rozwiązywania NP-trudnych problemów optymalizacji. Do pełnego zrozumienia i stosowania wymagają jednak gruntownej wiedzy z zakresu teorii statystyki matematycznej i prawdopodobieństwa.

3.4. Programowanie genetyczne

Programowanie genetyczne jest stosunkowo nowym kierunkiem w rozwoju algorytmów ewolucyjnych. W algorytmie tym zarówno dane jak i program reprezentowane są w postaci drzew. Programowanie genetyczne wykorzystujące język programowania LISP znajduje zastosowanie między innymi do automatycznego pisania programów komputerowych. Chromosomy mają charakterystyczną budowę, więc w odniesieniu do operatorów genetycznych również zastosowano odpowiednie modyfikacje.

3.5. Zastosowanie algorytmów ewolucyjnych

Algorytmy ewolucyjne znalazły zastosowanie między innymi w marketingu i sprzedaży, produkcji i finansach (Biethahn, Nissen 1995; Pierreval, Tautou 1997; Alander 1999;

Vergara i in. 2002), a także w harmonogramowaniu i planowaniu produkcji (Murata i in. 1996; Pawlak 1999; Knosala i zespół 2002; Serafin 2006). W zagadnieniach związanych z górnictwem, zastosowanie znalazły głównie algorytmy genetyczne m.in. do harmonogramowania produkcji (Pendharkar, Rodger 2000; Sequence optimization 2007), do doboru sprzętu i maszyn (Naoum, Haidar 2000) oraz szacowania niezawodności urządzeń (Vayenas, Yuriy 2007), a także do lokalizacji zabudowy kopalni (Kumral 2004) i poszukiwania optymalnej wydajności zakładu przerobczego (Mahajan 2004). Natomiast pewną koncepcję podejścia ewolucyjnego do modelowania robót górniczych przedstawiono w (Brzychczy 2006).

Właściwości, które przemawiają za wykorzystywaniem algorytmów ewolucyjnych do rozwiązywania złożonych zagadnień to (w odróżnieniu od algorytmów klasycznych):

- przyjęta reprezentacja, odpowiednia dla zadania,
- przestrzeń poszukiwań jest przeszukiwana z różnych punktów startu,
- rozwiązania konkurują ze sobą, co skutkuje osiągnięciem osobników o coraz to lepszym przystosowaniu,
- możliwość zmiany funkcji oceny podczas procesu ewolucyjnego, bez zatrzymania algorytmu,
- możliwość zrównoleżenia procesu ewolucyjnego w populacji.

Zaprezentowany w niniejszym artykule model jest pewną uproszczoną częścią zagadnienia dotyczącego modelowania wyposażenia robót eksploatacyjnych w wielozakładowym przedsiębiorstwie górniczym. Mnogość wariantów oraz czasochłonność obliczeń wymagają wprowadzenia nowoczesnych metod analizy i poszukiwania optymalnych rozwiązań podjętego problemu badawczego. Stąd też – uwzględniając charakterystyczne właściwości podejścia ewolucyjnego i jego niewątpliwe zalety – w Katedrze Ekonomiki i Zarządzania w Przemyśle AGH rozpoczęto prace nad konstrukcją odpowiedniego algorytmu ewolucyjnego (działającego na rozwiązaniach pełnych) dla przedstawionego pokrótce w niniejszym artykule zagadnienia.

Podsumowanie

W niniejszym artykule przedstawiono zagadnienie modelowania wyposażenia robót eksploatacyjnych w grupie kopalń, dla którego poszukiwana jest efektywna metoda rozwiązania. Zaprezentowano krótki przegląd metod klasycznych umożliwiających analizę sformułowanego problemu zarówno w postaci rozwiązań pełnych jak i częściowych z uwzględnieniem ich zalet i wad, do których odniesiono złożoność przedstawionego modelu.

W efekcie rozważań zwrócono uwagę na niewątpliwe korzyści i znaczne możliwości wykorzystania w tym zakresie podejścia ewolucyjnego, co jest obecnie przedmiotem prac badawczych autorki.

LITERATURA

- Alander J.T., 1999 – An Indexed Bibliography of Genetic Algorithm Implementations. Report Series No. 94-1-IMPLE, <ftp.uwasa.fi/cs/report94-1/galMPLEbib.ps.Z>.
- Arabas J., 2004 – Wykłady z algorytmów ewolucyjnych. Wydawnictwa Naukowo-Techniczne, Warszawa.
- Badania operacyjne. Praca zbiorowa pod red. E. Ignasiaka. PWE, Warszawa 1996.
- Biethahn J., Nissen V., 1995 – Evolutionary algorithms in management applications. Springer-Verlag, Berlin.
- Brzychczy E., 2006 – Zastosowanie algorytmów ewolucyjnych w modelowaniu produkcji górniczej. Zeszyty Naukowe Politechniki Śląskiej nr 1736, seria Górnictwo z. 273, Gliwice.
- Brzychczy E., 2007 – Propozycja modułu eksperckiego dla potrzeb systemu projektowania robót górniczych w kopalniach węgla kamiennego. Zeszyty Naukowe Politechniki Śląskiej nr 1764, seria Górnictwo z. 279, Gliwice.
- Gerstenkorn T., Śródka T., 1976 – Kombinatoryka i rachunek prawdopodobieństwa. PWN, Warszawa.
- Goldberg D.E., 2003 – Algorytmy genetyczne i ich zastosowania. Wydawnictwa Naukowo-Techniczne, Warszawa.
- Grabowski W., 1982 – Programowanie matematyczne. PWE, Warszawa.
- Klank M., 2002 – Racjonalizacja wykorzystania produkcyjnych środków trwałych w procesie restrukturyzacji górniczej grupy kapitałowej. Praca doktorska. Akademia Górniczo-Hutnicza, Kraków.
- Knosala i zespół, 2002 – Zastosowania metod sztucznej inteligencji w inżynierii produkcji. Wydawnictwa Naukowo-Techniczne, Warszawa.
- Kumral M., 2004 – Optimal location of a mine facility by genetic algorithms. Mining Technology: IMM Transactions section A, 113, 2.
- Mahajan A.M., 2004 – Performance optimization of a coal preparation plant using genetic algorithms. Final Technical Report 03-1/4.1B-4. Southern Illinois University.
- Michalewicz Z., Fogel D.B., 2006 – Jak to rozwiązać czyli nowoczesna heurystyka. WNT, Warszawa.
- Murata T., Ishibuchi H., Tanaka H., 1996 – Multi-objective genetic algorithm and its application to flowshop scheduling. Computers and Industrial Engineering, 30.
- Naoum S., Haidar A., 2000 – A hybrid knowledge base system and genetic algorithms for equipment selection. Engineering, Construction and Architectural Management, 7.
- Pawlak M., 1999 – Algorytmy ewolucyjne jako narzędzie harmonogramowania produkcji. Wydawnictwo Naukowe PWN, Warszawa.
- Pendharkar P.C., Rodger J.A., 2000 – Nonlinear programming and genetic search application for production scheduling in coal mines. Annals of Operations Research, 95.
- Pierreval H., Tautou L., 1997 – Using evolutionary algorithms and simulation for the optimization of manufacturing systems. IIIE Transactions, 29.
- Sequence optimization - www.mirarco.org/projects/cmt.php z dnia 10.07.2007.
- Serafin R., 2007 – Zastosowanie algorytmów ewolucyjnych do harmonogramowania zadań produkcyjnych. Komputerowo Zintegrowane Zarządzanie T. 1, pod red. R. Knosali, WNT, Warszawa.
- Vayenas N., Yurij G., 2007 – Using GenRel for reliability assessment of mining equipment. Journal of Quality in Maintenance Engineering, Emerald, Vol. 13, No.1.
- Vergara F.E., Khouja M., Michalewicz Z., 2002 – An evolutionary algorithm for optimizing material flow in supply chains. Computers and Industrial Engineering, 43.

TRADYCYJNE METODY OPTYMALIZACJI A NOWOCZESNA HEURYSTYKA W WYBRANYCH ZAGADNIENIACH
MODELOWANIA ROBÓT GÓRNICZYCH W KOPALNIACH WĘGLA KAMIENNEGO

Słowa kluczowe

Górnictwo węgla kamiennego, modelowanie, roboty górnicze, optymalizacja, heurystyka

Streszczenie

W artykule przedstawiono efekty poszukiwań odpowiedniej metody optymalizacji dla postawionego problemu badawczego. Rozważanym zagadnieniem z zakresu modelowania robót górniczych jest planowanie robót eksploatacyjnych w wielozakładowym przedsiębiorstwie górniczym. Należy zauważyć, że w warunkach obecnie funkcjonujących przedsiębiorstw, zakres tego planowania dotyczy możliwości wyposażenia planowanych robót, ponieważ parametry geometryczne pól ścianowych jak i kolejność ich wybierania są znacznie wyprzedzająco zapisywane i akceptowane w sporządzanych planach techniczno-ekonomicznych. Istota podjętego problemu oraz jego złożoność obliczeniowa wymagały poszukiwań efektywnej metody optymalizacji na zbiorze wielu możliwych rozwiązań. W artykule przedstawiono uproszczony model analizowanego zagadnienia: przyjętą reprezentację, cel oraz przykładową funkcję oceny rozwiązań. Z uwagi na objętość wzorów dla wyprowadzenia postaci funkcji oceny krótko przedstawiono najważniejsze zależności pomiędzy elementami składowymi funkcji a wartością postępu zawartą w reprezentacji problemu oraz, w pewnym uproszczeniu, tok generowania rozwiązania. W dalszej części publikacji przedstawiono charakterystykę wybranych klasycznych metod optymalizacji, które podzielono na algorytmy, oceniające tylko pełne rozwiązania oraz algorytmy, które wymagają oceny rozwiązań częściowo skonstruowanych (niepełne rozwiązania pierwotnego problemu) lub przybliżonych (pełne rozwiązanie zredukowanego problemu). Wskazano ich zalety i wady oraz odniesienie ich do sformułowanego problemu badawczego. Omówiono również podejście ewolucyjne, które umożliwia intensyfikację poszukiwań przestrzeni rozwiązań i tworzenie efektywnych heurystyk. Zaprezentowano kilka właściwości, które przemawiają za wykorzystywaniem algorytmów ewolucyjnych do rozwiązywania złożonych zagadnień, w tym również do przedstawionego w artykule przez autorkę.

TRADITIONAL METHODS OF OPTIMIZATION VS MODERN HEURISTIC IN MINING WORKS MODELING ISSUES IN HARD COAL MINES

Key words

Hard coal mining, modeling, mining works, optimization, heuristic

Abstract

Main issue displayed in the paper is the review of optimization methods according to formulated research problem such as planning of exploitation works in hard coal mining enterprise. It have to be stated that at the present time in polish hard coal mining, scope of the planning process is limited. Geometric parameters of longwall panels and its sequence of exploitation are rather determined in accepted long-term plans. So the main possibility is to analyze equipment for planned mining works. In presented article Author provided model of analyzed problem: assumed representation, aim and evaluation function. Take into consideration number of equations which are need to derive a formula of evaluation function, there were shortly presented main dependences between function component elements and assumed representation as well as simplified calculation procedure. Because of the problem complexity and time-consuming calculations Author undertook studies on search an effective optimization method to solve displayed research problem. In the article advantages and disadvantages of selected classic optimization methods as well as evolutionary approach were described. Classic methods were divided into: algorithms which estimate only complete solutions and algorithms for estimation of partial solutions (non complete solutions of original problem) or approximated solution (complete solution of reduced problem). In the article exhaustive search, local search and linear programming were presented as well as some greedy algorithms: dynamic programming, A* algorithm, divide and rule algorithm and divide and bound method. Simulated annealing and tabu search were presented also. As modern approach to search space of solutions evolutionary algorithms were described. Following the paper a few characteristic of evolutionary algorithms were presented which make this approach more applicab!e to considered problem than classic approach.