

# ROZWIJANIE KOMPETENCJI INFORMATYCZNYCH NA PRZYKŁADZIE PROBLEMU PLECAKOWEGO

*Herma Sławomir, Raczek Włodzimierz, Żywczak Bartłomiej*  
*V Liceum Ogólnokształcące, ul. Lompy 10*

*Akademia Techniczno-Humanistyczna ul. Willowa 2,*  
*43-300 Bielsko-Biała,*

[sławomir.herma@lo5.bielsko.pl](mailto:sławomir.herma@lo5.bielsko.pl), [sherma@ath.bielsko.pl](mailto:sherma@ath.bielsko.pl)  
[włodzimierz.raczek@lo5.bielsko.pl](mailto:włodzimierz.raczek@lo5.bielsko.pl) [bartłomiej.zywczak@lo5.bielsko.pl](mailto:bartłomiej.zywczak@lo5.bielsko.pl)

*Abstract. The paper presents the knapsack problem as an example of a problem with considerable teaching potential, giving a wide range of opportunities for multidirectional development of the student's IT competencies. The work is the result of several educational experiments carried out both by high school students and university students. The main intent of the Authors was to show the variety of ways of shaping and developing computational and algorithmic thinking at all educational stages.*

## 1. Wstęp

Pojęcie tzw. „potencjału dydaktycznego” odnośne jest zazwyczaj do jednostek akademickich, ich wydziałów lub katedr, najczęściej w rozumieniu zasobów ludzkich, którymi dysponują w zakresie realizowanych procesów kształcenia. Znacznie rzadziej pojęcie to używane jest w stosunku do jakiegoś, ogólnie pojmowanego, dzieła, choćby literackiego czy muzycznego, w kontekście jego walorów poznawczych i edukacyjnych. Czy można zatem rozszerzyć ten termin, określając nim również pewne zagadnienia, przykłady, sytuacje dydaktyczne projektowane przez nauczyciela po to, by skuteczniej, efektywniej, w możliwie atrakcyjny i przystępny sposób przekazać wiedzę uczniom lub studentom? Czy zasadne jest odnośzenie pojęcia „potencjału dydaktycznego” do zagadnień, które umiejętnie przedłużane, rozwijane, tworzą przestrzeń rozumienia nowych pojęć, pomagają pokonywać kolejne bariery abstrakcji, wzmacniają intuicje poznawcze, stymulują kreatywność, kształtują procesy myślowe, rozwijają kompetencje?

Osiąganie celów kształcenia, określonych w podstawie programowej, zależy w dużym stopniu od skutecznego zachęcania i aktywizowania uczniów do rozwiązywania wartościowych przykładów, zagadnień i tematów, właśnie takich, które

nauczycielowi dają szeroki wachlarz możliwości w zakresie stosowania wielorakich metod i środków dydaktycznych, zaś uczniowi sprawiających satysfakcję z osiągniętego sukcesu. Autorzy niniejszego artykułu, stawiając sobie za cel poszukiwanie interesujących zagadnień o tak rozumianym potencjale dydaktycznym, podjęli próbę adaptacji tzw. **problemu plecakowego** do realizacji zajęć na różnych etapach edukacyjnych, zależnie od przyjętej metody jego rozwiązania. W pracy zawarto więc propozycje kilku scenariuszy, kształtujących wybrane kompetencje informatyczne m.in:

- myślenie komputacyjne, i algorytmiczne,
- programowanie z użyciem wybranych środowisk i języków,
- zespołowe rozwiązywanie problemów.

## 2. Problem plecakowy – definicja

Problem plecakowy narodził się w środowiskach militarnych w okresie II Wojny Światowej, kiedy to podjęto działania zmierzające do optymalizacji zawartości plecaka żołnierza w zależności frontu, na który był wysyłany. Wyposażenie plecaka należało tak dobierać, by zmaksymalizować jego użyteczność (przydatność) dla danych warunków frontowych. Sam problem optymalizacyjny wynika więc z faktu, iż każdy element ze zbioru przedmiotów potencjalnych (tych, które poprzez odpowiednie przyporządkowanie mogą stanowić wyposażenie plecaka) charakteryzuje się pewną wartością użytkową, określającą jego przydatność na danym froncie, przy czym liczba tych elementów jest znacznie większa niż plecak mógłby pomieścić. Ponadto, chcąc zaopatrzyć żołnierza najlepiej jak to możliwe, trzeba być świadomym ograniczeń wynikających z dopuszczalnego udźwigu, którego po całkowitym załadunku nie wolno przekroczyć.

Całość zagadnienia sprowadza się zatem do problemu optymalizacyjnego, w którym celem nadrzędnym jest taki dobór wyposażenia, by elementy w plecaku były jak najbardziej przydatne (tzn. miały jak największą sumaryczną wartość użytkową), by znalazło się tam możliwie jak najwięcej różnorodnych elementów, i by nie przekroczyć dopuszczalnego udźwigu. Niekiedy rozważana jest jeszcze objętość plecaka, a wraz z nią – warunek nieprzekroczenia sumarycznej objętości zawartych w przestrzeni ładunkowej obiektów.

## 3. Programowanie bez komputera

### 3.1. Idea rozwiązania, zarys scenariusza

#### Przygotowanie zajęć

Nauczyciel przedstawia zagadnienie w postaci historyjki o smoku strzegącym wielkiego skarbu (pochodzącego z okradania dobrych ludzi), a uczniowie wcielają

się w sprytnych bohaterów, którzy chcą choćby część tego skarbu odzyskać, mając plecaki o określonej objętości i udźwigu. Smok codziennie ma w zwyczaju uciąć sobie krótką drzemkę, w czasie której każdy może niepostrzeżenie wejść do skarbcza i zebrać część zgromadzonych skarbów. Uczniowie muszą zatem zdecydować, które z nich warto zapakować do plecaka, by jego zawartość okazała się najbardziej wartościowa (tzn. by spośród utraconych rzeczy odzyskać jak najwięcej tych najcenniejszych), ale też była możliwa do udźwignięcia.

Wyobrażeniem skarbów mogą być pudełka o różnych wielkościach i ciężarze, a reprezentacją plecaka – pudło odpowiednio większe. W tym scenariuszu pudełka mają kształt prostopadłościenny i przygotowywane są samodzielnie przez uczniów w ramach zajęć plastycznych. Pomalowane są na różne kolory, a informacja o wartości użytkowej (w dolarach, złotych, punktach) napisana jest na jednej ze ścian. Na pudełku napisany jest również jego ciężar, który został wcześniej określony na zwykłej wadze np. szalkowej lub elektronicznej. Zawartość poszczególnych pudełek stanowią kamyki, kasztany, żołędzie, ziarna grochu, fasoli, kasza itp. zbierane lub przynoszone przez dzieci. Obiekty te mogą reprezentować np. złote monety, szlachetne kamienie albo też wartości wyższe, jak miłość, radość, pokój, uprzejmość, wiedza, mądrość, życzliwość, dobroć, wspinałomyślność, skromność oraz pozostałe wartości, jak: pieniądze, samochody, ładne ubrania, fajne gadzety.

Jedną z odmian powyższego pomysłu jest zastąpienie pudełek niewielkimi woreczkami z podobną zawartością. Przygotowanie zajęć staje się nieco łatwiejsze, a w momencie wprowadzenia pojęcia objętości, woreczki, w przeciwieństwie do prostopadłościennych pudełek, dokładniej wypełniają przestrzeń plecaka.

### Przebieg zajęć

- Nauczyciel dzieli uczniów na zespoły. Każdy zespół otrzymuje „plecak”, którym jest pudło (lub rzeczywisty plecak) umieszczony na wadze sprężynowej lub elektronicznej. Jeśli takowej nie ma, każdy zespół otrzymuje liczydło lub soroban (może być również kalkulator prosty) służące do kontroli (sumowania) wagi elementów. Kontrola objętości może odbywać się w każdym przypadku w formie wizualnej, doświadczalnej lub podobnie, jak w przypadku ciężaru – za pomocą sorobana czy też kalkulatora prostego.
- Uczniowie dysponując zbiorem przedmiotów – skarbów, wspólnie podejmują decyzje o umieszczeniu lub nie, określonego obiektu w przestrzeni plecaka.
- Zazwyczaj uczniowie nie mają większych trudności z oceną wartości elementów i intuicyjnie rozumieją, że w pierwszej kolejności należy wybierać te, które zostały uznane za najcenniejsze dla nich samych.
- W miarę podejmowania kolejnych decyzji uczniowie orientują się, że w plecaku zmieści się znacznie więcej elementów, gdy będą zwracać uwagę

na te o małym ciężarze i jednocześnie dużej wartości (ponieważ im więcej elementów znajduje się w plecaku tym jest on bardziej wartościowy). Gdy w ślad za tym nauczyciel zasugeruje uwzględnianie wielkości (tzn. objętości) obiektów w taki sposób, by i ją uczniowie minimalizowali, praca w zespołach staje się bardzo ożywiona, a element rywalizacji aktywizuje nawet tych uczniów, którzy do tej pory byli niezbyt zaangażowani w zabawę.

- Właśnie wprowadzenie przez nauczyciela pierwiastka ludycznego do tak zorganizowanych zajęć, w znacznym stopniu przyczynia się do uatrakcyjnienia ich przebiegu, kierując je w stronę nauki przez zabawę.

### Ocena uzyskanych rozwiązań i wnioski

Po przeprowadzeniu głównej części zajęć, uczniowie wraz z nauczycielem dokonują oceny uzyskanych rozwiązań przez poszczególne zespoły, wybierając to, którego sumaryczna wartość okazała się największa. W zależności od etapu edukacyjnego, na którym przeprowadzono zajęcia, można dodatkowo poprosić uczniów by spróbowali zwerbalizować metodę przyjętą w takcie rozwiązywania zadania. Uczniom bardziej ambitnym można pozostawić sformułowanie rozwiązania w postaci listy kroków (czyli algorytmu).

## 3.2. Odniesienia do wybranych kompetencji informatycznych

W rozumieniu nowej podstawy programowej dla szkół podstawowych z dnia 14 lutego 2017 r, w zakresie edukacji informatycznej, realizacja powyższego scenariusza może pozytywnie wpływać na uzyskiwanie osiągnięć w zakresie m.in:

- rozumienia, analizowania i rozwiązywania problemów: (układania w logicznym porządku obrazów, tekstów, poleceń <instrukcji>, tworzenia poleceń lub sekwencji poleceń dla określonego planu działania prowadzących do osiągnięcia celu, rozwiązywania zadań i łamigłówek prowadzących do odkrywania algorytmów),
- rozwijania kompetencji społecznych (m.in. współpracy z uczniami, wzajemnej wymiany pomysłów i doświadczeń).

Warto zauważyć, że zaproponowany scenariusz rozwija również szereg kompetencji spoza obszaru informatyki.

## 4. Programowanie zachłanne

### 4.1. Idea rozwiązania

#### Dane wejściowe

Dany jest zbiór elementów, które mają być użyte do wyposażenia plecaka;

$$\Omega = \{\omega_n\}_{n=1,\dots,N}$$

oraz jednokolumnowe tablice (wektory), określające odpowiednio: ciężary i objętości poszczególnych elementów z zbioru  $\Omega$ ;

$$C = [c_n]_{n=1,\dots,N}, \quad O = [o_n]_{n=1,\dots,N}$$

Ponadto dany jest wektor, określający wartość użytkową poszczególnych elementów ze zbioru  $\Omega$ ;

$$W = [w_n]_{n=1,\dots,N}$$

oraz udźwig plecaka  $c_0$  i jego całkowitą objętość  $o_0$ ;

Przyjmuje się, że każdemu elementowi  $\omega_n$  odpowiada ciężar  $c_n$ , tzn.  $\omega_n \hat{=} c_n$ , objętość tzn.  $\omega_n \hat{=} o_n$  oraz pewna wartość użytkowa  $w_n$ , tzn.  $\omega_n \hat{=} w_n$ .

### Ograniczenia

- zbiór  $\Omega'$  elementów przyporządkowanych do plecaka musi zawierać się w zbiorze  $\Omega$  - wszystkich elementów, tzn.:  $\Omega' \subset \Omega$ .
- suma ciężarów przyporządkowanych do plecaka elementów nie może przekroczyć dopuszczalnego udźwigu plecaka  $c_0$  tzn.  $\sum_{\omega_n \in \Omega'} c_n \leq c_0$ ;
- suma objętość przyporządkowanych do plecaka elementów nie może przekroczyć całkowitej jego objętości  $o_0$  tzn.  $\sum_{\omega_n \in \Omega'} o_n \leq o_0$ ;

### Kryterium optymalizacji

Kryterium optymalizacji polega na maksymalizacji wartości użytkowej (sumarycznej) elementów przyporządkowanych do plecaka tzn.

$$Q = \sum_{\omega_n \in \Omega'} w_n \rightarrow \max.$$

### Heurystyki

Heurystyka jest procedurą (metodą postępowania), która w pewnym stopniu odzwierciedla procesy myślowe człowieka. Największą jej zaletą jest szybkość

działania. Otrzymane rozwiązanie jest dopuszczalne i zbliżone do optymalnego. Na potrzeby rozpatrywanego problemu definiowane są następujące heurystyki:

$$1) \quad H_1 : \forall_{n=1, \dots, N} \left( \frac{1}{c_n} \right) \rightarrow \max.$$

Dla każdego elementu  $\omega_n \in \Omega$  obliczany jest iloraz  $\frac{1}{c_n}$ . Otrzymane wyniki porządkowane są w kolejności malejącej. Do plecaka przydzielane są elementy o największym ilorazie. Postępowanie prowadzi się do momentu, w którym dodanie kolejnego elementu spowoduje przekroczenie dopuszczalnego udźwigu plecaka. Słowem, pobiera się kolejne elementy od najlżejszych do najcięższych.

$$2) \quad H_2 : \forall_{n=1, \dots, N} (w_n) \rightarrow \max.$$

Sposób postępowania w przypadku heurystyki  $H_2$  jest analogiczny, jak przy poprzedniej. Uzyskuje się tutaj efekt maksymalizacji wartości użytkowej przyporządkowywanych elementów bez zwracania uwagi na ich ciężary jednostkowe. Postępowanie prowadzi się do momentu, w którym dodanie kolejnego elementu spowoduje przekroczenie dopuszczalnego udźwigu plecaka.

$$3) \quad H_3 : \forall_{n=1, \dots, N} \left( \frac{w_n}{c_n} \right) \rightarrow \max$$

Heurystyka ta uwzględnia zarówno ciężar jednostkowy elementu, jak i jego wartość użytkową. W wyniku jej działania w plecaku znajdują się elementy, posiadające największą wartość użytkową przy jednoczesnym minimalnym ciężarze.

Wobec faktu dodatkowego uwzględnienia objętości, można rozpatrywać ponadto:

$$4) \quad H_4 : \forall_{n=1, \dots, N} \left( \frac{1}{o_n} \right) \rightarrow \max.$$

$$5) \quad H_5 : \forall_{n=1, \dots, N} \left( \frac{1}{c_n \cdot o_n} \right) \rightarrow \max.$$

oraz:

$$6) \quad H_6 : \forall_{n=1, \dots, N} \left( \frac{w_n}{c_n \cdot o_n} \right) \rightarrow \max .$$

### Komentarz

Powyższy model matematyczny, przedstawia w sposób sformalizowany ideę programowania zachłannego, korzystając z aparatu matematycznego niedostępnego jeszcze dla uczniów szkół podstawowych, gimnazjów czy niektórych liceów lub techników, lecz znanego nauczycielom informatyki i matematyki. Mogą więc oni w każdej chwili zaadoptować go do potrzeb uczniów, kierując się własną intuicją, w zależności od danego etapu edukacyjnego.

## 4.2. Propozycja realizacji zajęć

Ideę programowania zachłannego można pokazać w bardzo przyjazny i łatwy dla ucznia sposób, wykorzystując arkusz kalkulacyjny np. MS Excel (na II etapie edukacyjnym według nowej podstawy programowej z 2017 r. – tj. w klasach IV-VIII), korzystając z przykładowego, ramowego scenariusza:

- Nauczyciel wprowadza zagadnienie plecakowe, krótko wyjaśniając podłoże historyczne problemu.
- Uczniowie tworzą prosty arkusz, zbudowany z kilku kolumn (w pierwszej znajdują się numery elementów lub ich konkretne nazwy, w drugiej – ich wartości użytkowe podane w umownie przyjętych jednostkach, w trzeciej ciężary, w czwartej objętości). Aby ograniczyć żmudne wprowadzanie danych wejściowych do arkusza, nauczyciel może zasugerować użycie funkcji LOS lub LOS.ZAKR (MS Excel) służących do generowania liczb pseudolosowych.
- Z tak przygotowanego zestawu danych należy utworzyć tabelę (Ctrl+T, MS Excel), by ułatwić późniejsze ich szeregowanie. Symulację załadunku plecaka uczniowie wykonują po uporządkowaniu elementów wg ich wartości użytkowych, ciężarów lub objętości. Korzystając z wbudowanych mechanizmów arkusza (np. sum zwykłych i warunkowych), kontrolują ciężar plecaka i stopień jego wypełnienia, by nie przekroczyć obu dopuszczalnych wartości.
- Nauczyciel pobudza intuicję uczniów proponując rozszerzenie tabeli o kolejne kolumny zawierające obliczone ilorazy według podanych wcześniej heurystyk oraz sugeruje wykonanie kolejnych symulacji i porównanie ich wyników. Bardzo istotne w zrozumieniu zagadnienia oraz uzyskaniu pozytywnych efektów kształcenia jest ponadto skorzystanie z gotowych narzędzi, automatyzujących tworzenie wykresów i ułatwiających tym samym wizuali-

zając podejmowanych przez uczniów decyzji podczas symulowanego pakowania plecaka.

Zagadnienie plecakowe stanowi również ciekawą propozycję w zakresie realizacji zajęć wg innego, przykładowego, ramowego scenariusza, który wykorzystuje programowanie w wybranym języku np. C++ (dotyczy obowiązującej podstawy programowej dla szkół ponadgimnazjalnych). W tym wypadku:

- Uczniowie, po zapoznaniu się z charakterystyką zagadnienia, rozważają model algorytmu dla przyszłej aplikacji, zwracając uwagę na dobór odpowiednich struktur danych (w tym typów tablicowych) niezbędnych do realizacji zadania.
- Ponieważ model zakłada konieczność szeregowania elementów według różnych wskaźników (ilorazów), istotny staje się dobór odpowiednich algorytmów porządkowania (metodą bąbelkową, przez scalanie lub inną).
- Podział klasy na grupy zadaniowe, z których każda programuje zagadnienie odmienną metodą, daje możliwość porównania uzyskanych wyników przy założeniu takich samych zestawów danych wejściowych.

Opisane tu scenariusze były zrealizowane w ramach przygotowań, a następnie prowadzenia warsztatów naukowych w ramach XVIII Beskidzkiego Festiwalu Nauki i Sztuki. W trakcie tych warsztatów, przeznaczonych dla licealistów oraz studentów, uczestnicy znajdowali niekiedy rozwiązania lepsze od tych, które uzyskiwane były przy pomocy zaprogramowanych uprzednio heurystyk w arkuszu MS Excel. Zaktywizowani, poprzez wprowadzenie pierwiastka rywalizacji, uczestnicy intuicyjnie odkryli algorytm Horowitza-Sahni-ego, będący również heurystyką dla problemu plecakowego.

### 4.3. Odniesienia do wybranych kompetencji informatycznych

Rozwiązanie zagadnienia plecakowe metodą programowania zachłannego pozytywnie wpływa na rozwijanie m.in.:

- umiejętności sprawnego posługiwania się arkuszem kalkulacyjnym do modelowania i symulacji prostych i bardziej złożonych zagadnień algorytmicznych, korzystania z dostępnych narzędzi w tym graficznych, wizualizujących uzyskiwane rozwiązania, pracy zespołowej, odkrywania algorytmów,
- umiejętności dostrzegania ścisłych zależności pomiędzy specyfikacją problemu w postaci modelu matematycznego a doбором właściwych struktur danych przy projektowaniu algorytmów,
- umiejętności integrowania różnych rozwiązań algorytmicznych w ramach jednej aplikacji;



## 5. Programowanie wieloetapowe

### 5.1. Idea rozwiązania

#### Model procesu decyzyjnego

Dane wejściowe, warunki brzegowe (ograniczenia) oraz kryterium optymalizacji zostały przyjęte analogicznie jak w opisie modelu matematycznego dla algorytmu zachłannego (p. 4.1), dlatego zostaną tu pominięte.

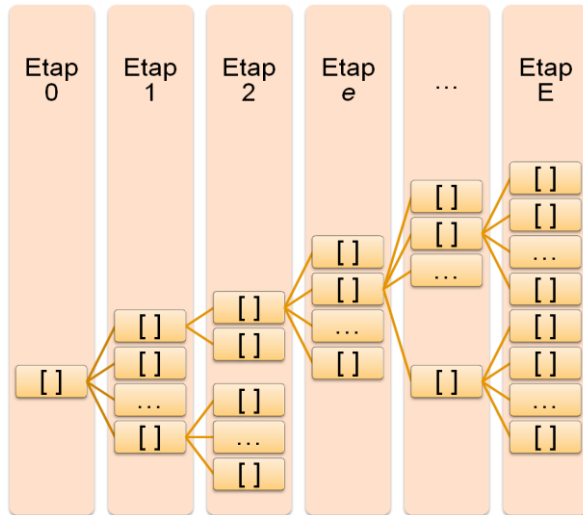
Zgodnie z propozycją przedstawioną przez Mareckiego w [1], do rozwiązania opisywanego zagadnienia przyjęto model programowania wieloetapowego, polegający na permutacyjnym generowaniu przyporządkowań elementów  $\omega_n$  i wyborze optymalnego spośród wszystkich uzyskanych rozwiązań (kierując się wspomnianym już kryterium). Realizacja tego zadania wymaga zdefiniowania pojęcia stanu procesu decyzyjnego. Skoro ostatecznym rezultatem opisywanego zagadnienia ma być optymalnie zapakowany plecak, zasadne (i wygodne z punktu widzenia dalszej implementacji informatycznej algorytmu) wydaje się, by stan procesu odzwierciedlony był za pomocą jednowymiarowej struktury tablicowej (wektora), zawierającej wartości „1” dla każdego przydzielonego elementu  $\omega_n$  lub „0” gdy elementu nie przydzielono.

Przykład:

$$P = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{array}{l} \rightarrow \omega_1 \text{ nie ma w plecaku} \\ \rightarrow \omega_2 \text{ jest w plecaku} \\ \rightarrow \omega_3 \text{ jest w plecaku} \\ \rightarrow \omega_4 \text{ nie ma w plecaku} \\ \rightarrow \omega_5 \text{ nie ma w plecaku} \\ \rightarrow \omega_6 \text{ jest w plecaku} \\ \rightarrow \omega_7 \text{ nie ma w plecaku} \end{array}$$

Idea algorytmu programowania wieloetapowego została przedstawiona w opracowaniach [5] oraz [8]. W opisywanym przypadku, schemat decyzyjny rozłożony jest na kilka etapów. Etap „0” – początkowy, charakteryzuje się tym, że wektor stanu zawiera wyłącznie wartości „0”. Oznacza to, że plecak jest pusty i nie zawiera żadnych elementów. Kolejny etap „1” zawiera zestaw wektorów stanu, z których każdy posiada już przypisany jeden i tylko jeden, jednak za każdym razem inny, element. Liczba stanów na tym etapie jest równa liczbie elementów ze zbioru  $\Omega$ . Na każdym kolejnym etapie procesu decyzyjnego podejmowana jest próba tworze-

nia zestawu wektorów stanu z kolejnym przypisanym do harmonogramu elementem  $\omega_n$  (spełniającym warunki brzegowe). Schemat tego procesu jest drzewem decyzyjnym, które, jako byt hierarchiczny, ma na każdym kolejnym etapie wektory stanu, dziedziczące niejako uzyskane już na etapach poprzednich decyzje. Stąd też, etap końcowy zawierać będzie zestaw rozwiązań problemu, czyli harmonogramów realizujących wszystkie możliwe, dopuszczalne przyporządkowania (patrz Rysunek 1).



**Rysunek 1.** Ogólny schemat generowania ścieżek decyzyjnych w algorytmie programowania wieloetapowego.

Zatem matematyczna definicja wektora stanu przyjmuje postać:

$$P^{e,l} = [p_i^{e,l}]_{i=1,\dots,N}$$

gdzie:

$$p_i^{e,l} = \begin{cases} 1 \rightarrow \text{przedmiot } \omega_i \text{ jest w plecaku} \\ 0 \rightarrow \text{w przeciwnym wypadku} \end{cases}$$

W szczególności, stan początkowy jest wektorem zerowym; wszystkie jego elementy  $p_i$  mają wartość 0.

### Procedura generowania stanu

Kolejny stan procesu decyzyjnego można wygenerować dla każdego elementu  $\omega_n$  ze zbioru  $\Omega$  (wszystkich dostępnych elementów), jeśli w stanie poprzednim

$P^{e-1,\lambda}$  na pozycji odpowiadającej  $n$ -temu elementowi była wartość „0” (tzn.  $n$ -ty element nie był jeszcze dotychczas przyporządkowany -  $p_n^{e-1,\lambda} = 0$ ). Ponadto wartość stanu  $V_1^{e-1,\lambda}$  (rozumiana jako suma ciężarów dotychczas przyporządkowanych elementów) powiększona o ciężar elementu „ $n$ ” nie może przekroczyć udźwigu dopuszczalnego  $c_0$  tj. ( $V_1^{e-1,\lambda} + c_n \leq c_0$ ) oraz wartość stanu  $V_2^{e-1,\lambda}$  (rozumiana jako suma objętości dotychczas przyporządkowanych elementów) powiększona o objętość elementu „ $n$ ” nie może przekroczyć całkowitej objętości plecaka  $o_0$  ( $V_2^{e-1,\lambda} + o_n \leq o_0$ ). Wówczas stan kolejny obliczany jest jako suma wektora stanu poprzedniego i wektora  $\Delta p$  tj. ( $P^{e,l} = P^{e-1,\lambda} + \Delta p$ ). Wektor  $\Delta p$  charakteryzuje się tym, że na każdej pozycji ma liczbę „0” z wyjątkiem tej, która odpowiada elementowi przyporządkowywanemu -  $\omega_n$ .

$$\forall_{n=1,\dots,N} \left[ \left( p_n^{e-1,\lambda} = 0 \right) \wedge \left( V_1^{e-1,\lambda} + c_n \leq c_0 \right) \wedge \left( V_2^{e-1,\lambda} + o_n \leq o_0 \right) \right] \Rightarrow P^{e,l} = P^{e-1,\lambda} + \Delta p$$

gdzie:

$$\Delta p_i = \begin{cases} 1 \rightarrow i = n \\ 0 \rightarrow \text{w przeciwnym wypadku} \end{cases}$$

Przykładowa realizacja przyporządkowania elementu  $\omega_3$  do plecaka:

$$\omega_3 \rightarrow \begin{matrix} P^{e-1,\lambda} \\ \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{matrix} + \begin{matrix} \Delta p \\ \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \end{matrix} = \begin{matrix} P^{e,l} \\ \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \end{matrix}$$

### Definicja wartości stanu

Wartością stanu  $P^{e,l}$  nazywa się parametr  $V^{e,l}$ , pozwalający wyselekcjonować spośród wszystkich wygenerowanych w procesie decyzyjnym stanów ten, który jest optymalny z punktu widzenia przyjętego kryterium. Zdefiniowano go jako sumę wartości użytkowych wszystkich przyporządkowanych w danym stanie elementów do plecaka:

$$V^{e,l} = \sum_{i=1}^N p_i^{e,l} \cdot w_i$$

## 5.2. Propozycja realizacji zajęć

Realizację powyższego zagadnienia wg zaproponowanego modelu matematycznego realizowane są przez Autorów w ramach zajęć uzupełniających, dla uczniów liceum w klasach z rozszerzeniem mat. - inf. oraz dla studentów na kierunkach informatycznych. Zazwyczaj, przy rozwiązywaniu tego zadania korzysta się z języka VBA (w środowisku MS Excel) oraz C++ lub C# (CodeBlocks, Microsoft Visual Studio). W tym przypadku, wymagania względem uczestników zajęć sięgają już znajomości wybranych algorytmów programowania dynamicznego. Ponadto przyjmuje się tu formułę pracy zespołowej nad wspólnym projektem lub indywidualnej – dla uczniów wybitnie uzdolnionych.

## 5.3. Odniesienia do wybranych kompetencji informatycznych

Do wymienionych uprzednio efektów w zakresie rozwijania wybranych kompetencji informatycznych, należy dołączyć:

- rozwijanie i utrwalanie umiejętności rozwiązywania wybranych zagadnień za pomocą różnych klas algorytmów oraz oceny ich efektywności i przydatności w badanej dziedzinie,
- kształtowanie umiejętności łączenia różnych rozwiązań algorytmicznych celem budowy złożonych aplikacji w różnych środowiskach programistycznych,
- pogłębianie i utrwalanie wiedzy na temat złożoności obliczeniowej i czasowej algorytmów, zwłaszcza w kontekście rozwiązywania zagadnień NP-trudnych, należących do grupy problemów decyzyjnych,
- rozwijanie umiejętności posługiwania się językiem sformalizowanym w celu precyzyjnego opisu modeli wybranych algorytmów,
- dostrzeganie szerszego wachlarza zastosowań wybranych algorytmów, nie tylko w dziedzinie informatyki i matematyki.

## 6. Podsumowanie

Problem plecakowy ilustruje znacznie więcej interesujących zagadnień, niż sugerowałaby sama jego nazwa. Wiele spośród zadań ekonomicznych, inżynierskich czy też przemysłowych można sformułować jako zagadnienie plecakowe:

- zagadnienia załadunku (np. ciężarówki, wagonu kolejowego czy innej przestrzeni ładunkowej w tym przestrzeni dla materiałów płynnych) [3, 5],

- zagadnienia rozcinań (np. prętów o standaryzowanej długości, tafli szkła o znormalizowanych wymiarach, celem minimalizacji uzyskiwanych odpadów lub maksymalizacji ciętych elementów), [2 ,4]
- zagadnienia ochrony danych (np. szyfr plecakowy),
- zagadnienia wyboru projektów, kontroli budżetu i lokaty kapitału [6].

Wprowadzenie opisywanego zagadnienia do procesu dydaktycznego wydaje się sprzyjać rozwojowi szeregu nie tylko informatycznych kompetencji ucznia i jednocześnie wpisywać skuteczną realizację wspólnych celów kształcenia dla wszystkich etapów edukacji.

## Literatura

1. Bucki R., Marecki F., *Digital Simulation of Discrete Processes*, Network Integrators Associates, 2006.
2. Christofides N., Mingozzi A., Toth P., Scandi C.: *Combinatorial Optimization*, Wiley, New York 1979.
3. Eilon E., Christofides N.: *The loading problems*, Management Sci. 17, 1971.
4. Gilmore P.C., Gomory R.E.: *Theory and computation of knapsack functions*, Oper. Res. 14, 1966.
5. Sysło M.M., Deo N., Kowalik J.S.: *Algorytmy optymalizacji dyskretnej*, PWN, Warszawa 1999.
6. Weingartner H.: *Mathematical Programming and the Analysis of Capital Budgeting Problems*. Prentice – Hall, Englewood Cliffs, NJ 1963.