

## NOWOCZESNE NARZĘDZIA OPTYMALIZACJI – laboratorium 2

### Wybrane elementy modeli programowania z ograniczeniami (CP-SAT)

#### *Ograniczenia logiczne*

1. Utwórz model programowania z ograniczeniami weryfikujący formuły logiczne zapisane w koniunkcyjnej postaci normalnej (CNF) dla czterech zmiennych i czterech klauzul. Korzystając z modelu utwórz i zapisz w dokumentacji dwie przykładowe formuły spełnialne wraz z przykładowymi wartościami spełniających je zmiennych.
2. Opracuj program, który najpierw wygeneruje i wypisze losową formułę logiczną CNF, następnie zweryfikuje jej spełnialność, a na końcu wyświetli wynik (wartości zmiennych lub informację o niespełnialności). Program powinien korzystać z parametrów:  $N$  – liczba zmiennych,  $C$  – liczba klauzul,  $K_{min}/K_{max}$  – minimalna/maksymalna liczba zmiennych w klauzuli. Wygeneruj i zamieść w dokumentacji dwie przykładowe formuły, jedną spełnialną i jedną niespełnialną, uzyskane dla  $N = 4$ ,  $C = 20$ ,  $K_{min} = 2$ ,  $K_{max} = 4$ .

#### *Dopasowanie zasobów*

3. Opracuj model programowania z ograniczeniami, który pozwoli zoptymalizować dopasowanie gości do pokoi hotelowych na podstawie preferowanych elementów wyposażenia. Zaproponuj dane opisujące co najmniej 8 pokoi i 8 gości oraz 3 elementy wyposażenia. W implementacji zastosuj ograniczenie indeksowe oraz „wszystkie różne”. Czy taki model będzie poprawny również w przypadku różnej liczby pokoi i gości? Odpowiedź uzasadnij.
4. Rozbuduj program z punktu poprzedniego tak, aby pozwolił na wczytywanie danych z plików. W jednym pliku powinny być zawarte dane o pokojach, a w drugim o klientach. Zaproponuj takie formaty plików, aby można było stosować zmienne liczby pokoi, klientów i elementów wyposażenia. Przetestuj utworzony program.

#### *Problem komiwojażera*

5. Wybierz na płaszczyźnie 5 różnych punktów  $A, \dots, E$  o współrzędnych  $x, y \in \{0, 1, \dots, 6\}$ . Wyznacz odległości między każdą parą punktów w macierzy miejskiej. Utwórz model programowania z ograniczeniami, który pozwoli wyznaczyć najkrótszy cykl biegnący dokładnie raz przez wszystkie punkty (cykl Hamiltona). W implementacji zastosuj ograniczenie *circuit*. Program optymalizacyjny powinien wypisać uzyskaną minimalną drogę oraz przebieg trasy od punktu  $A$ , np.  $A \rightarrow C \rightarrow B \rightarrow D \rightarrow E$ .
6. Rozbuduj program z punktu poprzedniego tak, aby wczytywał dane z pliku w postaci macierzy odległości. Liczba punktów powinna być rozpoznawana na podstawie zawartości pliku. Przebieg trasy ma być wypisany za pomocą numerów punktów (począwszy od 1), zamiast liter, jak poprzednio. Przetestuj implementację.