

Problem

W winiarzy (oznaczonych dalej przez W_i) produkuje każdy po X_i litrów wina.

S studentów (oznaczonych dalej przez S_j) konsumuje każdy po Y_j litrów wina.

Aby przekazać wino, winiarz musi wynająć bezpieczne miejsce B_k .

Winiarz nie rozpocznie produkcji wina, dopóki nie odda wszystkiego, co już wyprodukował.

Przy założeniach:

$$i \in \{1, 2, \dots, W\}$$

$$j \in \{1, 2, \dots, S\}$$

$$k \in \{1, 2, \dots, N\}$$

$$\neg(\forall_{i \in \{1, 2, \dots, W\}} \exists_{j \in \{1, 2, \dots, S\}} X_i = Y_j)$$

Proponowane rozwiązanie

Aby rozwiązać podany problem musimy poczynić dodatkowe założenia:

- $\sum X_i = \sum Y_j$
- Winiarz nie musi oddać całej partii jednemu studentowi ($X_i \geq Y_j$)
- Student nie musi zaspokoić wszystkich swoich potrzeb u jednego winiarza ($Y_j \geq X_i$)
- Student może w razie potrzeby zaspokoić tylko część swojego zapotrzebowania, jednak pozostała część musi zostać zaspokojona tak szybko jak to tylko możliwe.

Założenie pierwsze zapobiega problemowi nadprodukcji. Jeśli założymy że proces jest ciągły i nieskończony, to przybiera ono formę:

$$\lim_{n \rightarrow \infty} n \times \sum X_i = \lim_{n \rightarrow \infty} n \times \sum Y_j$$

i jest niespełnialne w praktyce.

Założenia drugie i trzecie zapobiegają sytuacji, w której popyt i podaż sumarycznie się równoważą, ale niemożliwy jest przydział całościowy.

Założenie czwarte jest potrzebne w przypadku procesu ciągłego i nieskończonego, pozwalając na złamanie założenia pierwszego w czasie jednej iteracji zakładając, że zostanie ono skorygowane w czasie następnych iteracji (zbyt mała podaż w i -tej iteracji zostanie zrównoważona nadpodażą w $i + 1$ iteracji)

Opis algorytmu

1. Winiarz W_i "produkuje" X_i litrów wina i ubiega się o bezpieczne miejsce B_i (sekcja krytyczna).
2. Po otrzymaniu B_i , W_i rozsyła informację o X_i do wszystkich studentów.
3. Student S_j określa swoje zapotrzebowanie na Y_j litrów wina i ubiega się o dostęp do winiarzy (sekcja krytyczna).
4. Jeśli dostępne są zasoby wina, S_j otrzymuje dostęp (zgodnie z algorytmem przydziału), w przeciwnym przypadku oczekuje na wiadomość od W_i .
5. Po otrzymaniu dostępu do winiarzy S_j , wybiera winiarzy od których weźmie wino
6. Po wyborze winiarzy S_j odbiera wino i (jeśli odebrał całą partię) zwalnia B_i i informuje o tym W_i .
7. Jeśli zapotrzebowanie S_j nie może zostać spełnione, to rozgłasza on wyczerpanie zasobów wina i ponownie ubiega się o dostęp do winiarzy dostając najwyższy priorytet.