Implementation of a Hard Random Instances Generator for the Open Shop Scheduling Problem

Salah Eddine Bouterfif

June 11, 2021

In their effort to calculate a new lower bound for the open shop problem, [1] showed that most open shop instances produced by random generators are easy and proposed a hard instances generator.

The random generator takes the number of jobs n and the number of machines m as input and requires three parameters k, p and f.

We have adapted the generator from [1] to produce rectangle open shop instances. Algorithm 1 describes how an instance is generated in two steps. The first step creates the $n \times m$ matrix P of processing times using the parameter k such that all elements of P are equal to $\frac{k}{m}$ (rounded down to the nearest integer). The second step performs a number of perturbations p to the matrix P.

A parameter f is used to calculate v: a fixed value *to be subtracted* and mv: a *maximum subtractable* value.

The parameters k, p and f are described by:

- k: integer number, such that the sum of each line of P is equal to k. The value $k \mod m$ is added to the diagonal of P.

- p: number of perturbations, such that for each perturbation, two task's processing times P_{ij} and P_{kl}, $i \neq k$ and $j \neq l$, are randomly selected from P, from which is subtracted the value of v.

- $f = \{f \in \mathbb{R} \mid 0 < f < 1\}$, such that:

  $mv = \min\{P_{ij}, P_{kl}\} - 1$ (subtract 1 to avoid tasks of length 0).

  $v = (f \cdot mv) + r$, with r: a random value between 0 and $f \cdot mv$.

The value subtracted from P_{ij} and P_{kl} is added to P_{il} and P_{kj} to keep the sum of all rows equal to k.

Algorithm 1 HOSSP: Hard Random Instance Generator.

procedure HOSSP(n, m, K, p, f)
$\triangleright$ Step 1: create a matrix P
  $inc := 0$
  if $n = m$ **then** $\triangleright$ square problems
    for each $i \leq n$ and $j \leq m$ **do**
      $P_{i,j} := \dfrac{K}{m}$
      $P_{i,i} := P_{i,i} + K \mod m$
    end for each
  else $\triangleright$ rectangle problems
    $m' = \min\{n, m\}$
    for each $i \leq n$ and $j \leq m$ **do**
      if $inc = m'$ **then** $inc := 0$
      end if
      $P_{i,j} := \dfrac{K}{m}$
      if $i \geq m'$ and $j = inc$ **then**
        $P_{i,j} := P_{i,j} + K \mod m$
      end if
      if $i < m'$ **then**
        $P_{i,i} := P_{i,i} + K \mod m$
      end if
      $inc := inc + 1$;
    end for each
  end if
$\triangleright$ Step 2: Add perturbations to the matrix P
  for each $i \leq p$ **do**

    while $i \neq j$ and $k \neq l$ **do**
      Randomly select $P_{i,j}, P_{k,l}$
    end while

    $mv = \min\{P_{ij}, P_{kl}\} - 1$ $\triangleright$ maximum subtractable value
    $r := random(0, mv)$ $\triangleright$ random value to subtract
    $v = (f \cdot mv) + r$ $\triangleright$ total value to subtract

    $P_{i,j} := P_{i,j} - v$
    $P_{k,l} := P_{i,j} - v$

    $P_{i,l} := P_{i,l} + v$
    $P_{k,j} := P_{k,j} + v$

  end for each

end procedure

For numerous generated instances the parameters k, p and f are randomly generated as follows:

- k: random value between $n \times m$ and $n \times m \times 100$
- p: random value between $n \times m$ and $n \times m^2$
- f: random value between 0 and 1

In the case of $n \neq m$, $k \mod m$ is added the diagonal of top-down square sub-matrices of size $\min\{n, m\}$ of P. The main diagonal is used on the remaining rectangle ensuring the sum of all rows of P are equal to k, this ensure that .

References

[1] Guéret, C., & Prins, C. (1999). A new lower bound for the open-shop problem. Annals of Operations Research, 92, 165–183. https://doi.org/10.1023/A:1018930613891.