DM811 – Fall 2009
Heuristics for Combinatorial Optimization

Lecture 9
Local Search

Marco Chiarandini

Deptartment of Mathematics & Computer Science
University of Southern Denmark

Outline

1. Random Numbers

2. LS Application Examples

Random Numbers

Carachtersitics of a good pseudo-random generator
(from stochastic simulation)

- long period
- uniform unbiased distribution
- uncorrelated (time series analysis)
- efficient

Suggested: MRG32k3a by L'Ecuyer
http://www.iro.umontreal.ca/~lecuyer/

```
java.lang.Object
  extended by umontreal.iro.lecuyer.rng.RandomStreamBase
      extended by umontreal.iro.lecuyer.rng.MRG32k3a
```


Ideal Random Shuffle

Let's consider a sequence of n elements: $\{e_1, e_2, \dots e_n\}$.
The ideal random shuffle is a permutation chosen uniformly at random from the set of all possible $n!$ permutations.

- π_1 is uniformly randomly chosen among $\{e_1, e_2, \dots e_n\}$.
- π_2 is uniformly randomly chosen among $\{e_1, e_2, \dots e_n\} - \{\pi_1\}$.
- π_3 is uniformly randomly chosen among $\{e_1, e_2, \dots e_n\} - \{\pi_1, \pi_2\}$
- ...

Joint probability of $(\pi_1, \pi_2 \dots \pi_n)$ is $\frac{1}{n} \cdot \frac{1}{n-1} \cdot \dots 1 = \frac{1}{n!}$

```
long int* Random::generate_random_array(const int& size) {
   long int i, j, help;
   long int *v = new long int[size];
   for ( i = 0 ; i < size; i++ )
     v[i] = i;
   for ( i = 0 ; i < size-1 ; i++) {
     j = (long int) ( ranU01( ) * (size - i));
     help = v[i];
     v[i] = v[i+j];
     v[i+j] = help;
   }
   return v; }
```

Outline

1. Random Numbers

2. LS Application Examples

The Single Machine Total Tardiness Problem

Given: a set of n jobs $\{J_1, \ldots, J_n\}$ to be processed on a single machine and for each job J_i a processing time p_i, a weight w_i and a due date d_i.

Task: Find a schedule that minimizes
the total weighted tardiness $\sum_{i=1}^{n} w_i \cdot T_i$
where $T_i = \max\{C_i - d_i, 0\}$ (C_i completion time of job J_i)

Example:

Job	J_1	J_2	J_3	J_4	J_5	J_6
Processing Time	3	2	2	3	4	3
Due date	6	13	4	9	7	17
Weight	2	3	1	5	1	2

Sequence $\phi = J_3, J_1, J_5, J_4, J_1, J_6$

Job	J_3	J_1	J_5	J_4	J_1	J_6
C_i	2	5	9	12	14	17
T_i	0	0	2	3	1	0
$w_i \cdot T_i$	0	0	2	15	3	0

Scheduling in Parallel Machines

Definition

Total Weighted Completion Time on Unrelated Parallel Machines Problem
Input: A set of jobs J to be processed on a set of parallel machines M. Each job $j \in J$ has a weight w_j and processing time p_{ij} that depends on the machine $i \in M$ on which it is processed.

Task: Find a schedule of the jobs on the machines such that the sum of weighted completion time of the jobs is minimal.

Steiner Tree

Input: A graph $G = (V, E)$, a weight function $\omega : E \mapsto \mathbf{N}$, and a subset $U \subseteq V$.

Task: Find a Steiner tree, that is, a subtree $T = (V_T, E_T)$ of G that includes all the vertices of U and such that the sum of the weights of the edges in the subtree is minimal.

Vertices in U are the special vertices and
vertices in $S = V \setminus U$ are Steiner vertices.