DM811

Heuristics for Combinatorial Optimization

Lecture 11
Examples

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

1. Recap.

2. Examples
   - TSP
   - Graph Coloring
   - Indirect Solution Representation

Outline

Summary: Local Search Algorithms
(as in [Hoos, Stützle, 2005])

For given problem instance π:

1. search space S_π
2. neighborhood relation $\mathcal{N}_\pi \subseteq S_\pi \times S_\pi$
3. evaluation function $f_\pi : S \to \mathbf{R}$
4. set of memory states M_π
5. initialization function $\texttt{init} : \emptyset \to S_\pi \times M_\pi)$
6. step function $\texttt{step} : S_\pi \times M_\pi \to S_\pi \times M_\pi$
7. termination predicate $\texttt{terminate} : S_\pi \times M_\pi \to \{\top, \bot\}$

After implementation and test of the above components, improvements in efficiency (ie, computation time) can be achieved by:

A. fast delta evaluation

B. neighborhood pruning

C. clever use of data structures

Improvements in quality can be achieved by:

D. application of a metaheuristic

E. definition of a larger neighborhood

Outline

Outline

Asymmetric TSP into Symmetric TSP

How to encode an asymmetric TSP into a symmetric TSP?

Outline

Local Search for Graph coloring

Different choices for the candidate solutions, neighborhood structures and evaluation function define different approaches to the problem

k-fixed	complete	proper	
k-fixed	partial	proper	$+++$
k-fixed	complete	unproper	$+++$
k-fixed	partial	unproper	$-$
k-variable	complete	proper	$++$
k-variable	partial	proper	$-$
k-variable	complete	unproper	$++$
k-variable	partial	unproper	$-$

Preprocessing rules

Polynomial time simplifications

k-coloring (k fixed)

- Remove under-constrained nodes
- Remove subsumed nodes
- Merge nodes that must have the same color

Outline

1. Recap.

2. Examples
   - TSP
   - Graph Coloring
   - Indirect Solution Representation

Scheduling in Parallel Machines

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

Steiner Tree Problem

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.