A Meta-Greedy Approach applied to a Multiobjective Scheduling Problem

Louis-Claude CANON, Emmanuel JEANNOT

Project-Team AlGorille
Loria/INRIA/Université Henri Poincaré
Scheduling for large-scale systems, Knoxville, France

May 13, 2009
1. A meta-greedy approach
2. Example: a scheduling problem
3. Some preliminary results
4. Conclusion
Outline

1. A meta-greedy approach
2. Example: a scheduling problem
3. Some preliminary results
4. Conclusion
Motivation

Multiobjective problem

Need to handle antagonist objectives. Solutions can be incomparable (non-dominated).

Techniques

Fast method for generating a set of non-dominated solutions (possibly Pareto-optimal).
A meta-greedy approach

Beyond heuristics

General methodology
Method for solving multiobjective problems: takes a problem as input and produces a heuristic. Similar to multiobjective metaheuristic and greedy strategy. Restriction to the input problems: solutions can be constructed incrementally (as for greedy).

More precisely
Generalization of greedy algorithms when dealing with multiple objectives.
A meta-greedy approach

Multiobjective

Classical greedy

Any following incremental modification to a partial solution is chosen according to one criterion $c_1$ (from red to green).

Considering multiple objectives

A set of non-dominated solutions is constructed at each step.
Each incremented solution (red) is considered at the next iteration:

for each iteration
  for each solution in the population
    increment the solution in several ways
  keep the best generated partial solutions
Criteria specification

Required problem-specific specification

Similar to mutation and crossover operators for metaheuristic. Since each partial solutions need to be evaluated, criteria for comparing partial solutions need to be defined.

Remark

Intermediate criteria \(\neq\) final criteria.
How to compare partial solutions in a \textit{fair} way (good intermediate criteria)?
Outline

1. A meta-greedy approach

2. Example: a scheduling problem

3. Some preliminary results

4. Conclusion
Example: a scheduling problem

Problem definition

Workload and platform model

A parallel application consisting DAG, i.e. of a set of tasks with precedence constraints. A set of completely linked heterogeneous processors subject to failures.

Objectives

- obtain a feasible schedule (start and end times for each task)
- minimize the makespan (total duration) of the schedule
- minimize the failure probability of the schedule
Yet another algorithm based on HEFT

An order is defined for assigning task to processor. Tasks are iteratively assigned to the processor that minimizes a single criterion (end time of the current task).
Intermediate criteria definition
Remark: each partial solution has the same assigned tasks.
First criterion: makespan of the partial solution.
Second criterion: reliability.

Issue
Does not degenerate into the HEFT heuristic.
Indeed, partial makespan $\neq$ end time of the inserted task.
Idea: mix of both criteria: for equal makespans, solution with the lowest end time of the current task is better for time
Outline

1. A meta-greedy approach
2. Example: a scheduling problem
3. Some preliminary results
4. Conclusion

Some preliminary results

A meta-greedy approach

Example: a scheduling problem

Some preliminary results

Conclusion
### Other multiobjective approaches

<table>
<thead>
<tr>
<th>HEFT-sub</th>
<th>HEFT with a subset of processors</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEFT-agg</td>
<td>HEFT with aggregation</td>
</tr>
</tbody>
</table>

### Experiments

31 instances were generated (from Strassen, Gaussian elimination, and Cholesky decomposition). The hypervolume of the meta-greedy is better than HEFT-sub in 93% and always better than HEFT-agg.
Some preliminary results

Specific case

The graph is generated from the Strassen algorithm on a random platform.

![Pareto fronts graph]

- HEFT-sub
- HEFT-agg
- Meta-greedy

Failure proba vs Makespan
Set coverage indicator

Proportion of solutions in the column that are dominated by the row.
Outline

1. A meta-greedy approach
2. Example: a scheduling problem
3. Some preliminary results
4. Conclusion
Main contributions

- Propose a generic approach (on the same level as greedy and metaheuristic designs) that can be applied to many problems.
- Assess its efficiency on a scheduling problem.
- Raise principal issue: intermediate criteria selection (comparing partial solutions).

Perspective

- Use other task orderings (HSA, BSA) and other scheduling policies.
- Complete study of other combinatorial problems (knapsack, . . . ).
Conclusion

Set limitation size

Preference ordering

If too much generated solutions: selection among non-dominated solutions (active field of research).
Indicator-based proposition by Zitzler and Thiele (2009): keep a subset of solutions such that the indicator is maximized.
Parameters of the produced heuristics: indicator and maximum size.

The hypervolume

![Diagram of hypervolume with axes c1 and c2 and region R]