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

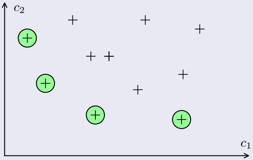
Outline

Outline

Motivation

Mutliobjective 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).
Existing method: multiobjective metaheuristics, espilon-method, Pareto set approximation (Papadimitriou, Yannakakis, 2000), ...

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).

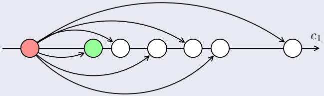
More precisely

Generalization of *greedy algorithms* when dealing with *multiple objectives*.

Multiobjective

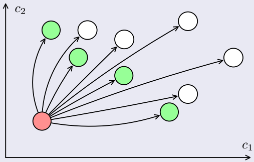
Classical greedy

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.

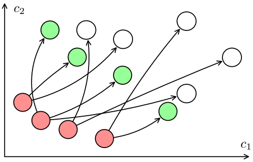


Main loop

Algorithm

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

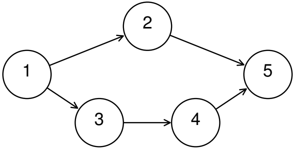
Intermediate criteria $\neq$ final criteria.
How to compare partial solutions in a *fair* way (good intermediate criteria)?

Outline

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.



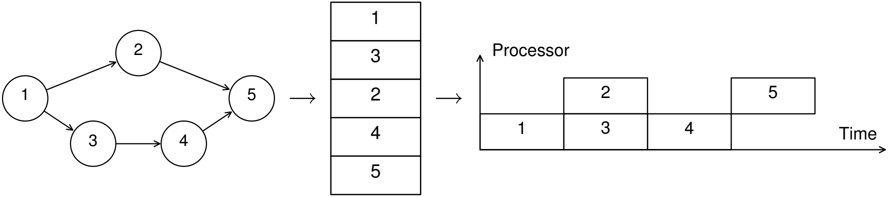
Objectives

- 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

Existing mono-objective greedy heuristics

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).



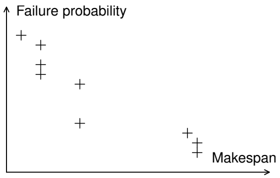
Meta-greedy implementation

Intermediate criteria definition

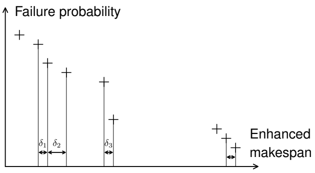
Remark: each partial solution has the same assigned tasks.
First criterion: makespan of the partial solution.
Second criterion: reliability.

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



$\longrightarrow$



Outline

Hypervolume indicator

Other multiobjective approaches

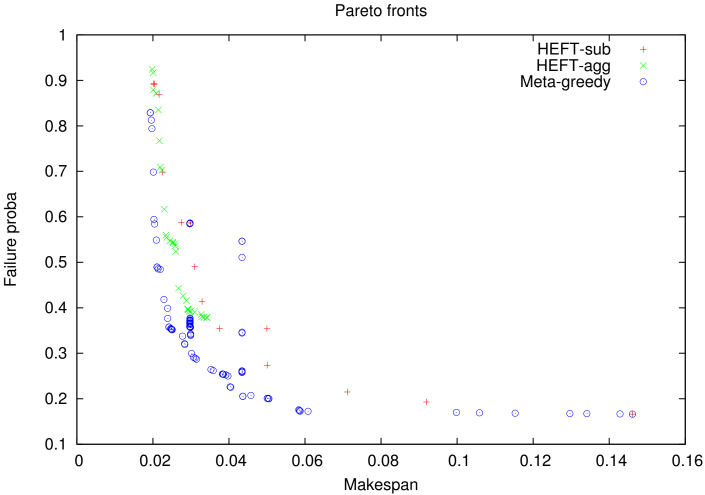
- **HEFT-sub** HEFT with a subset of processors
- **HEFT-agg** HEFT with aggregation

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.

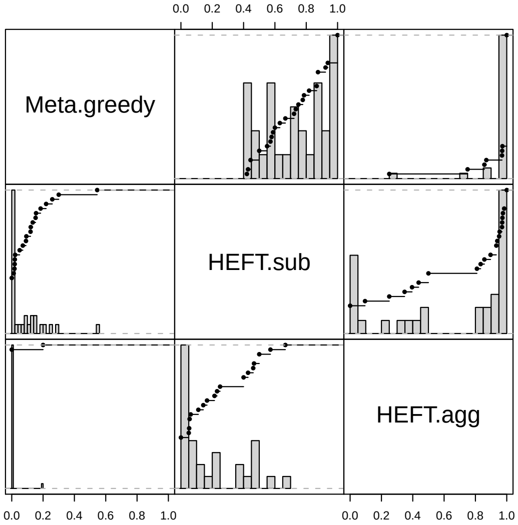
Specific case

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



Set coverage indicator

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



Outline

Conclusion and future directions

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, . . .).

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