Session 7 Discussion
Recorded by Yijun Yu, The Open University, UK

Guaranteeing Solution Quality for SAS Optimization Problems by being Greedy

Ulrike Stege from UVic.ca

Abstract: The talk is about greedy algorithms, using the Data Centre Scheduling problem as an example.

Given a set of jobs, distribute them on to a server.

More specifically, given a job, arrival time, deadline, process time and revenue as constraints, try to maximise the revenue.

// Marin: Deadline is a hard constraint for a job? Ulrike: Yes

The goals is to produce a fast solution, to produce good quality solution.

It defines greedy algorithms and indicates when it is good to use them for good quality solutions.

Greedy algorithm is not always the best.

Apply three layered policy solutions: action => goal => utility policies [POLICY 2004]

Decide on what is a better algorithm or better problem is the aim.

Generic maximisation problem defined

// John M: can constraints be logical formula? Yes, possible. F is a subset of U.

// Ric: are there other metrics? Yes there might be.

Specifically, { (job, starting_time) } is the universe, constrained by

arrival_time <= starting_time <= deadline - processing_time

Objective function is the sum of revenues in a given schedule.

objective function-based framework in literature

linear,

submodular

f (A U B) + f(A ^ B) <= f(A) + f(B)

// Cheerful sound from Kostas' laptop

constraint-based framework in literature

matroid

downward closed if A \subset B, B \in F then A \in F

why? Rick said: removing a job from the feasible schedule is still feasible

augmentation property

|A| < |B|, then exists x \in B \ A, A union {x} \in F

not always work

k-extensible

exchange property

A \subset B, exists x \in U - B, A \union {x} \in F,

then exists Y \subset B - A, | Y| <=k, (B-Y) \union {x} \in F

// Ric: what about an intermediate set C? It can help with the scheduling

// Hausi: Bashar do you have question? Bashar does not ask question, but smiles.

Example: does not satisfy the augmentation property, but satisfy the exchange properties

// Ulrike: Trust me: it works.

Conclusion: it is linear, but not matroid

// Tatsuo: k is not specified? Yes, k needs to be specified

// Zhenjiang: is k a constant? Yes.

S=0; A = 0;

repeat

A = { e | S \union {e} \in F}

if A != 0 then

u = argmax g(S \union {x}) | x \in A

S = S \union {u}

end if

until A = \phi

return S

Edmonds 71: matroid + linear => greedy is best

Fisher 78: matroid + objective => 1/2 approixmation

Mestre 06: k-extensible + linear => 1/k proximation

Fisher 78: k-exnteisble + modular => 1/(k+1) approximation

// JM: is 1/k approximation a lower bound? Hausi: yes, but it is a guarantee

add structure to the constraint set could get a guarantee

If processing time are unit, then it is optimal.

Three-step Recipe:

// Kostas: how does the recipe work? what about the overhead? You have to update the revenue with a trade-off function for the cost.

// Hausi: HP Lab has a report on how to formulate the objective function

// Tetsuo: complexity algorithm? it is often linear time

// Zhenjiang: Polynomial time.

Group activities

to formalise your optimisation problem into Scheduling?

// fun is not over...

case, how to recover?