Generating Training Data for Learning Linear Composite Dispatching Rules for Scheduling

Helga Ingimundardóttir
Thomas Philip Runarsson

University of Iceland

January 14, 2015

Outline

Introduction

Job Shop Scheduling

Preference Learning

Evolutionary search with CMA-ES

Experiments

Summary and Future Work

Introduction

Job Shop Scheduling

Preference Learning

Evolutionary search with CMA-ES

Experiments

Summary and Future Work

Motivation

General Goal

- General goal is how to search for *good* solutions for an arbitrary problem domain.
- Automate the design of optimization algorithms based on preference learning.
- Use of randomly sampled problem instances and their corresponding optimal vs. suboptimal solutions.

Case Study: JSP

Abstract

- Framework for creating dispatching rules for JSP.
- Linear classification to identify good dispatches from worse ones.
- Generate training data both from optimal and suboptimal solutions, by exploring various trajectories within the feature-space.
- Sample training data using different ranking schemes.

Keywords: Scheduling • Composite dispatching rules • JSP • Generating Training Data • Trajectory Sampling Strategies • Ranking Schemes

Introduction

Job Shop Scheduling

Preference Learning

Evolutionary search with CMA-ES

Experiments

Summary and Future Work

Job Shop Scheduling (1)

JSP

Simple job shop scheduling problem is where n jobs are scheduled on a set of m machines, subject to constraints:

- each job must follow a predefined machine order,
- that a machine can handle at most one job at a time.

Objective: schedule the jobs so as to minimize the maximum completion time, i.e., makespan, $C_{\max}$.

Job Shop Scheduling (2)

Problem space distributions used in experimental studies

	name	size	N_{train}	N_{test}	note
JSP	$\mathcal{P}_{j.rnd}^{6\times5}$	6×5	500	500	random
	$\mathcal{P}_{j.rndn}^{6\times5}$	6×5	500	500	random-narrow

Job Shop Scheduling (3)

Dispatching rules (DR) for constructing JSP

- Starts with an empty schedule and adds on one job at a time.
- When a machine is free the DR inspects the waiting/available jobs and selects the job with the highest priority.
- Complete schedule consists of $\ell = n \cdot m$ sequential dispatches.
- At each dispatch k, features $\phi(k)$ for the temporal schedule are calculated.

Performance of DR is compared with its optimal makespan is:

$$\rho = \frac{C_{\max}^{DR} - C_{\max}^{opt}}{C_{\max}^{opt}} \cdot 100\%$$

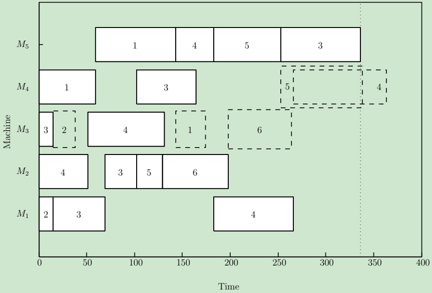
Job Shop Scheduling (4)

Features for JSP

ϕ	Feature description
ϕ_1	processing time for job on machine
ϕ_2	start-time
ϕ_3	end-time
ϕ_4	when machine is next free
ϕ_5	current makespan
ϕ_6	work remaining
ϕ_7	most work remaining
ϕ_8	slack time for this particular machine
ϕ_9	slack time for all machines
ϕ_{10}	slack time weighted w.r.t. number of operations already assigned
ϕ_{11}	time job had to wait
ϕ_{12}	size of slot created by assignment
ϕ_{13}	total processing time for job

Job Shop Scheduling (5)

Example

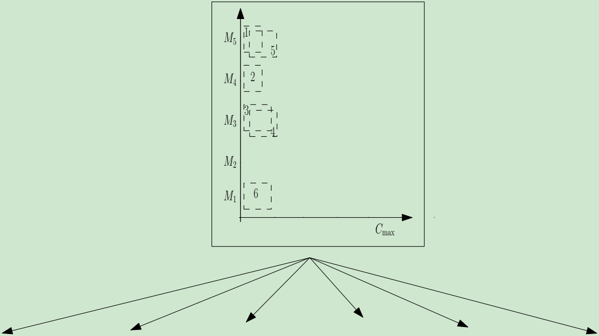


A schedule being built at step $k = 16$. The dashed boxes represent five different possible jobs that could be scheduled next using a DR.

Game-tree representation (1)

Example

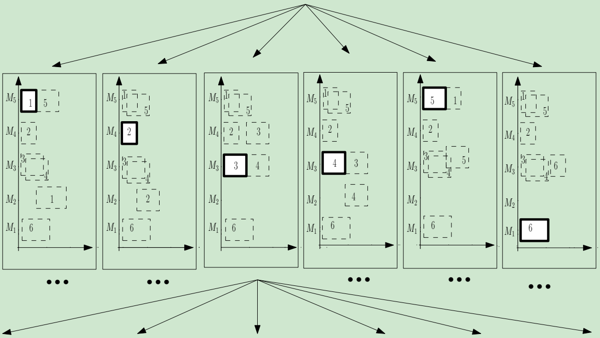
First layer (i.e. root) – empty schedule at step $k = 1$



Game-tree representation (2)

Example

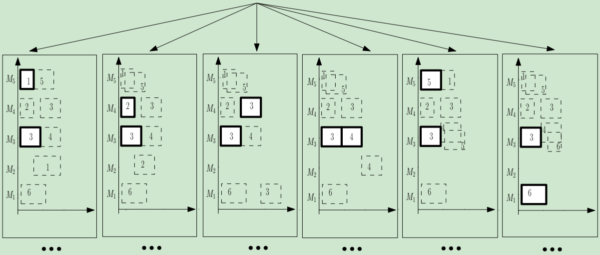
Second layer – all possible first dispatches at step $k = 2$



Game-tree representation (3)

Example

Third layer – given J_3 is dispatched first on M_3 at step $k = 3$



Introduction

Job Shop Scheduling

Preference Learning

Evolutionary search with CMA-ES

Experiments

Summary and Future Work

Ordinal Regression (1)

Preference learning problem

Specified by a set of preference pairs:

$$S = \left\{ \{\mathbf{z}_o, +1)\}_{k=1}^{\ell}, \{\mathbf{z}_s, -1)\}_{k=1}^{\ell} \mid \forall o \in \mathcal{O}^{(k)}, s \in \mathcal{S}^{(k)} \right\} \subset \Phi \times Y$$

where the set of point/rank pairs are:

- Optimal decision: $\mathbf{z_o} = \phi^{(o)} - \phi^{(s)}$, ranked $+1$
- Suboptimal decision: $\mathbf{z_s} = \phi^{(s)} - \phi^{(o)}$, ranked -1

and $\phi_o, \phi_s \in \Phi \subset \mathcal{F}$ are features from the collected training set Φ.

Ordinal Regression (2)

- Mapping of points to ranks: $\{h(\cdot) : \Phi \mapsto Y\}$ where

$$\phi_o \succ \phi_s \quad \Leftrightarrow \quad h(\phi_o) > h(\phi_s)$$

- The preference is defined by a linear function, i.e. PREF model:

$$h(\phi) = \sum_{i=1}^{d} w_i \phi = \langle w \cdot \phi \rangle.$$

- Logistic regression learns the optimal parameters **w** by solving:

$$\min_{\mathbf{w}} \quad \tfrac{1}{2}\langle w \cdot w \rangle + C \sum_{j=1}^{|S|} \log\left(1 + e^{-y_j \langle w \cdot z_j \rangle}\right)$$

Generating preference set S (1)

- At each dispatch k, a number of data pairs are created
- Separate data set for each dispatch, i.e., total of ℓ models.

Previous sampling approach

The strategy was to follow some single optimal job $j \in \mathcal{O}^{(k)}$, thus creating $|\mathcal{O}^{(k)}| \cdot |\mathcal{S}^{(k)}|$ feature pairs at each dispatch k, resulting in a training size of:

$$l' = \sum_{q=1}^{N_{\text{train}}} \left(\sum_{k=1}^{\ell} |\mathcal{O}^{(k)}| \cdot |\mathcal{S}^{(k)}| \right)$$

HÁSKÓLI ÍSLANDS
VERKFRÆÐI- OG NÁTTÚRUVÍSINDASVIÐ

Generating preference set S (2)

Trajectory sampling strategies explored for adding features to Φ

- Φ^{opt} follow some (random) optimal task
- Φ^{cma} follow the task corresponding to highest priority, computed with fixed weights $\mathbf{w}$, which were obtained by optimising with CMA-ES.
- Φ^{mwr} follow the SDR most work remaining (MWR).
- Φ^{rnd} follow some random task.
- Φ^{all} union of all of the above, i.e.,

$$\Phi^{all} = \Phi^{opt} \cup \Phi^{cma} \cup \Phi^{mwr} \cup \Phi^{rnd}$$

Generating preference set S (3)

Ranking schemes implemented for adding preference pairs to S

- S_b all opt rankings r_1 vs. all possible subopt rankings r_i, $i \in \{2, ..., n'\}$
- S_f full subsequent rankings, i.e., all combinations of r_i and r_{i+1} for all $i \in \{1, ..., n'\}$.
- S_p partial subsequent rankings, similar ot S_f except if there are more than one operation with the same ranking, only one is needed to be compared to subsequent rank, i.e., $S_p \subset S_f$.
- S_a union of all of the above, i.e.,

$$S_a = S_b \cup S_f \cup S_p$$

where $r_1 > r_2 > \cdots > r_{n'}$ $(n' \leq n)$ are the rankings of $\mathcal{R}^{(k)}$.

Introduction

Job Shop Scheduling

Preference Learning

Evolutionary search with CMA-ES

Experiments

Summary and Future Work

Evolutionary search

Instead of using logistic regression for to find the weights $\mathbf{w}$ for linear preference function:

$$h(\phi) = \sum_{i=1}^{d} w_i \phi = \langle w \cdot \phi \rangle.$$

Covariance Matrix Adaptation Evolution Strategy (CMA-ES), is applied directly on the objective function.

Benefit No need to collect training data beforehand.

Drawback Computationally expensive to evaluate $\mathbb{E}\left[C_{\max}\right]$

Introduction

Job Shop Scheduling

Preference Learning

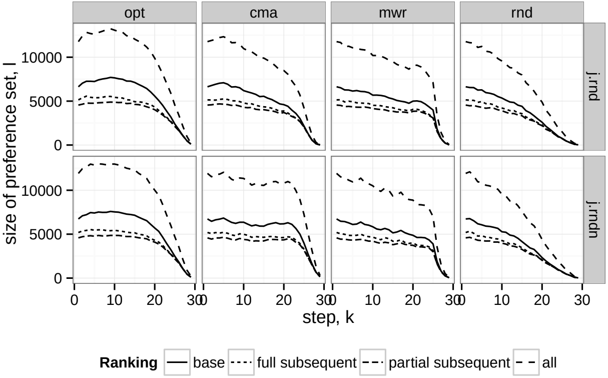
Evolutionary search with CMA-ES

Experiments

Summary and Future Work

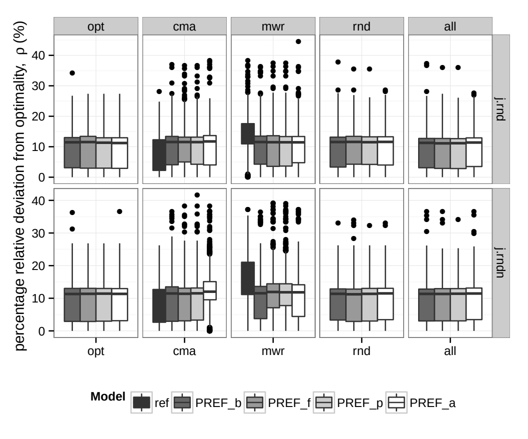
Experiments (1)

Size of preference set, $l = |S|$



Experiments (2)

Box-plot for PREF models using test set



Experiments (3)

Trajectory sampling strategies

- Learning preferences from good scheduling rules can be favourable.
- Tracking only optimal paths (Φ^{opt}) yield a generally lower mean relative error – although no statistical difference with Φ^{rnd}
- For $\mathcal{P}^{6\times5}_{j.rnd}$ the best model was based on Φ^{all} , where the suboptimal trajectories aid Φ^{opt} by adding a greater variety of preference pairs.

Results for ranking schemes

- No statistical difference between ranking schemes. However, opting for a smaller preference set then S_p is preferred.

Introduction

Job Shop Scheduling

Preference Learning

Evolutionary search with CMA-ES

Experiments

Summary and Future Work

Summary and future work (1)

- Introduced a framework for learning linear composite dispatch rules for scheduling based on preference learning.
- By partial subsequent ranking scheme it's possible to reduce the preference set without loss of performance.
- Success is highly dependent on the preference pairs introduced to the system, i.e., the trajectories explored through the feature-space.
- It is not obvious how to go about collecting training data.

Summary and future work (2)

- Learning optimal trajectories predominant in literature.
- In sequential decision making, all future observations are dependent on previous operations, so compound effect of errors can be dire.
- Study showed Φ^{opt} can result in insufficient knowledge of features.
- Learning from suboptimal schedules can improve the model when PREF^{opt} has diverged too far from Φ^{opt}.
- Limitations in linear approximation function to capture the complex dynamics incorporated in optimal trajectories.

Thank you for your attention

HÁSKÓLI ÍSLANDS
VERKFRÆÐI- OG NÁTTÚRUVÍSINDASVIÐ

Questions?

Contact: Helga Ingimundardóttir, `hei2@hi.is`