Automatic development of dispatching rules for the unrelated machines environment

Public defence of the thesis topic

Candidate: Marko Đurasević
Faculty of Electrical Engineering and Computing

Mentor: Prof. dr. sc. Domagoj Jakobović

October 2016

Outline

- Introduction
- The unrelated machines environment
- Creating dispatching rules with genetic programming
- Research objectives
  - DRs for multi-objective and many objective problems
  - Ensembles of DRs
  - DR selection based on problem instances
  - Scheduling in static environment

Introduction

- Scheduling – allocation of activities (jobs) to resources (machines) [1]
- Goal: optimise a given objective
- Motivation: NP-hard problem
- Many real world examples:
  - Scheduling in air traffic control
  - Scheduling in semiconductor manufacturing
  - Therapy scheduling in hospitals
  - Scheduling jobs in clusters

Unrelated machines scheduling

- *n* jobs which need to be scheduled on *m* machines
- Job parameters
  - Processing time (p_{ij})
  - Release time (r_j)
  - Due date (d_j)
  - Weight (w_j)
- Scheduling criteria
  - Makespan
  - Total weighted tardiness
- Dynamic and static conditions

Solving scheduling problems

- Exact algorithms
  - Dynamic programming, branch and bound

- Approximation algorithms

- Heuristic algorithms
  - Improvement heuristics
    - Genetic algorithms, tabu search
  - Constructive heuristics
    - Dispatching rules (DRs)

Solving scheduling problems with DRs

- Create schedule incrementally
  - A job is scheduled only when a machine is free
  - The order of the jobs is determined by using a priority function

- Examples of priority functions [3]:
  - EDD: $\frac{1}{d_j}$
  - WSPT: $\frac{w_j}{p_j}$
  - ERD: $\frac{1}{r_j}$

Solving scheduling problems with DRs

- Advantages
  - Complexity is negligible when compared to other approaches [2]
  - Can be used in dynamic and static environments
  - Can quickly react to changes in the scheduling environment

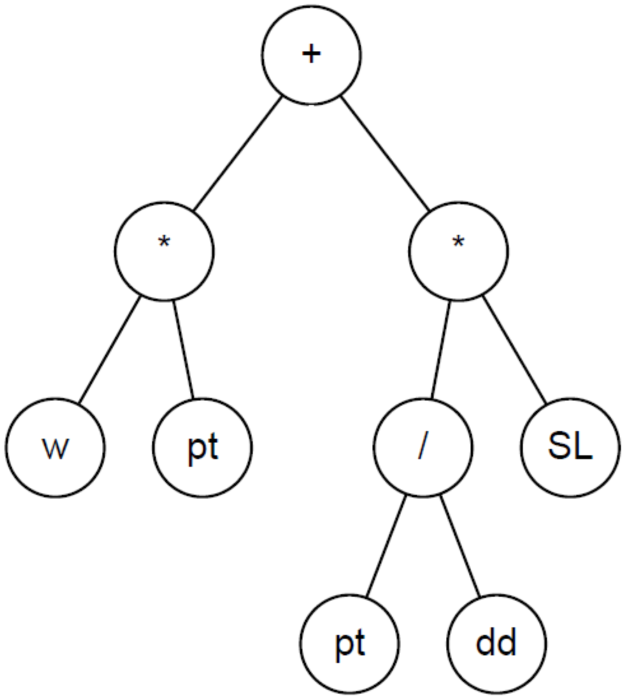
- Disadvantages
  - Can not create schedules of equal quality as other procedures
  - Difficult to design
  - Which DR to choose for a given problem instance

Creating DRs with GP

- Metaheuristic for solving optimisation problems by imitating the evolution process [4]

- Individuals represent mathematical functions and expression

- Evolves a priority function:
  - $(w * pt) + \frac{pt}{dd} * SL$



Creating DRs with GP

- Schedule generation scheme (SGS) – creates the schedule by using a priority function evolved by GP:
  - $(w * pt) + \frac{pt}{dd} * SL$

```
while (unscheduled jobs exist) {
     wait until a machine becomes ready;
     determine the priorities πi of all unscheduled jobs;
     schedule the job with the smallest/largest priority;
}
```

Creating DRs with GP

- Advantages
  - Easy to create new DRs for different scheduling criteria
  - Can outperform existing manually designed DRs [5]

- Open questions
  - Design of DRs for many-objective and multi-objective problems
  - Improving the performance of evolved DRs
  - Choosing the best DR to solve a scheduling problem
  - Improving performances in static environments
  - Interpretability of solutions
  - ....

DRs for multiobjective and many objective problems

- Many examples of multiobjective scheduling in real world
- DRs for multiobjective criteria are hard to design manually
- Not extensively covered in the literature
  - Reducing the multiobjective into a single objective problem [6]
  - Optimising five scheduling criteria by a multi objective GP [7] [8]

DRs for multiobjective and many objective problems

- Experiment with several criteria combinations and sizes

- Try out different multi-objective and many-objective procedures
  - NSGA-II
  - NSGA-III
  - HaD-MOEA
  - MOEA/D

- Determine the interdependences between different criteria

Ensembles of DRs

- Motivation: improve performances of DRs by ensemble learning methods

- Used in only one occasion
  - Only the Cooperative coevolution procedure [13]

- Use several procedures:
  - BoostGP [9]
  - BagGP [10] [11]
  - Cooperative coevolution [12]
  - Simple ensemble combination (proposed)

Ensembles of DRs

- Create ensembles of DRs used for scheduling
- Analysis of parameter influence
- Use an additional optimisation to reduce ensemble size and improve performances
- Testing for several optimisation criteria

DR selection based on problem instances

- Which of the evolved DRs should be used on an unseen problem instance?
- Determine the appropriate DR Based on the problem characteristics
- Some research already done in the literature [14] [15]
  - Only manually designed DRs are used
  - Mostly artificial neural networks are used for classification
  - No analysis parameter influence

DR selection based on problem instances

- Instead of manually designed DRs use evolved DRs

- Problem characteristics known in advance and approximated during execution

- Open questions
  - Which classification algorithms to use?
  - How many DRs to use?
  - How to approximate characteristics of problem instances?
  - How often should DRs be switched?
  - Influence of the learning set?
  - ....

Scheduling in static environment

- A need to build the schedule as fast as possible
- All parameters known in advance
- Improvement heuristics can be slow
- Not covered for the unrelated machines environment

Scheduling in static environment

- Determine additional information used by GP when evolving DRs

- Adjust the SGS

- Use different procedures
  - Iterative dispatching rules [16]
  - Rollout heuristic [17]

Conclusion

- DRs evolved by GP have shown to be able to outperform manually designed DRs
- Many topics still not adequately covered
- Possibility to significantly improve the results which can be achieved by the basic GP
- All approaches applicable in other machine environments and scheduling problems

Title and contibutions

- Automatiziran razvoj pravila raspoređivanja za okruženje nesrodnih strojeva
  1. Postupci raspoređivanja za višekriterijske i mnogokriterijske probleme raspoređivanja i analiza međusobne ovisnosti kriterija.
  2. Razvoj pravila raspoređivanja korištenjem metoda skupnog učenja.
  3. Postupak za odabir prilagodljivih pravila raspoređivanja s obzirom na svojstva instance problema.
  4. Razvoj pravila raspoređivanja za primjenu u statičkoj okolini raspoređivanja.
- Automatic development of dispatching rules for the unrelated machines environment
  1. Scheduling procedures for multi-objective and many-objective scheduling problems, and analysis of interdependence between criteria.
  2. Development of dispatching rules by the use of ensemble learning methods.
  3. Method for selection of adaptive dispatching rules based on the features of the problem instances.
  4. Development of dispatching rules for application in the static scheduling environment.

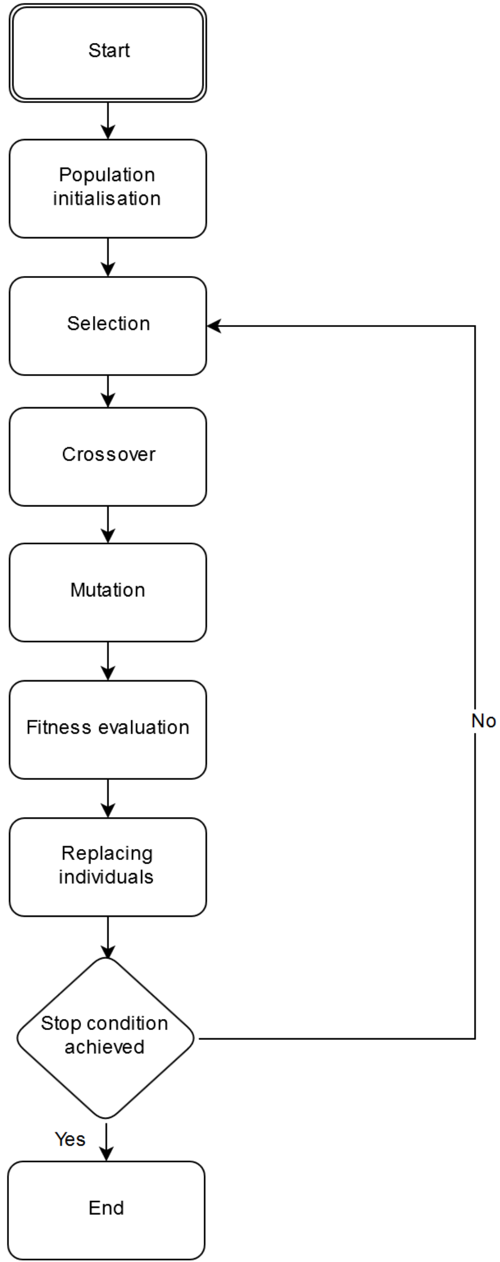
Title and contibutions - NEW

- Automatiziran razvoj pravila raspoređivanja za okruženje nesrodnih strojeva
  1. Postupak za razvoj pravila raspoređivanja za višekriterijske i mnogokriterijske probleme raspoređivanja temeljen na genetskom programiranju
  2. Primjena metoda skupnog učenja s ciljem poboljšanja kvalitete pravila raspoređivanja.
  3. Postupak za odabir prilagodljivih pravila raspoređivanja s obzirom na svojstva instance problema.
  4. Postupak razvoja pravila raspoređivanja primjenjivih u statičkoj okolini raspoređivanja.
- Automatic development of dispatching rules for the unrelated machines environment
  1. Procedure for development of dispatching rules for multi-objective and many-objective scheduling problems based on genetic programming
  2. Application of ensemble learning methods with the objective of improving the quality of dispatching rules.
  3. Method for selection of adaptive dispatching rules based on the features of the problem instances.
  4. Procedure for developing dispatching rules applicable in the static scheduling environment.

Example of an evolved DR

- $\pi = pos\left(\frac{pos(w*SL)}{\frac{w*pavg}{SL+pmin}}\right) - \left(pos\big((w*pt)\big) + (w*age) - \left(\left(\frac{pmin}{w}\right) - (MR - age) + (dd - pt) + (pmin + pavg)\right)\right.$

GP algorithm overview



Title of the thesis

- Automatic development of dispatching rules for the unrelated machines environment

- Automatiziran razvoj pravila raspoređivanja za okruženje nesrodnih strojeva

Contributions

1. Scheduling procedures for multi-objective and many-objective scheduling problems, and analysis of interdependence between criteria.

2. Development of dispatching rules by the use of ensemble learning methods.

3. Method for selection of adaptive dispatching rules based on the features of the problem instances.

4. Development of dispatching rules for application in the static scheduling environment.

Contributions

1. Postupci raspoređivanja za višekriterijske i mnogokriterijske probleme raspoređivanja i analiza međusobne ovisnosti kriterija.

2. Razvoj pravila raspoređivanja korištenjem metoda skupnog učenja.

3. Postupak za odabir prilagodljivih pravila raspoređivanja s obzirom na svojstva instance problema.

4. Razvoj pravila raspoređivanja za primjenu u statičkoj okolini raspoređivanja.

Contribution 1

- Scheduling procedures for multi-objective and many-objective scheduling problems, and analysis of interdependence between criteria.

- Postupci raspoređivanja za višekriterijske i mnogokriterijske probleme raspoređivanja i analiza međusobne ovisnosti kriterija.

Contribution 2

- Development of dispatching rules by the use of ensemble learning methods.

- Razvoj pravila raspoređivanja korištenjem metoda skupnog učenja.

Contribution 3

- Method for selection of adaptive dispatching rules based on the features of the problem instances.

- Postupak za odabir prilagodljivih pravila raspoređivanja s obzirom na svojstva instance problema.

Contribution 4

- Development of dispatching rules for application in the static scheduling environment.

- Razvoj pravila raspoređivanja za primjenu u statičkoj okolini raspoređivanja.

References

[1] M. L. Pinedo, Scheduling: theory, algorithms, and systems, Springer Science & Business Media, 2012.

[2] Đurasevic, M., Jakobovic, D.: Comparison of solution representations for scheduling in the unrelated machines environment

[3] Maheswaran, M., Ali, S., Siegal, H., Hensgen, D., Freund, R.: Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems. Proceedings. Eighth Heterogeneous Computing Workshop (HCW'99) (June 1999) (1999).

[4] Poli, R., Langdon, W.B., McPhee, N.F.: A field guide to genetic programming. Published via http://lulu.com and freely available at http://www.gp-field-guide.org.uk (2008). (With contributions by J. R. Koza)

[5] Đurasevic, M., Jakobovic, D., Kneževic, K.: Adaptive scheduling on unrelated machines with genetic programming. Applied Soft Computing **48**, 419 – 430 (2016).

[6] Tay, J.C., Ho, N.B.: Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. Computers and Industrial Engineering **54**(3), 453–473 (2008). DOI 10.1016/j.cie.2007.08.008

[7] Nguyen, S., Zhang, M., Tan, K.C.: Enhancing genetic programming based hyperheuristics for dynamic multi-objective job shop scheduling problems. 2015 IEEE Congress on Evolutionary Computation (CEC) (MAY), 2781–2788 (2015).

[8] Nguyen, S., Zhang, M., Johnston, M., Tan, K.C.: Dynamic multi-objective job shop scheduling: A genetic programming approach. In: A.S. Uyar, E. Ozcan, N. Urquhart (eds.) Automated Scheduling and Planning, *Studies in Computational Intelligence*, vol. 505, pp. 251–282. Springer (2013)

[9] Iba, H.: Bagging, boosting, and bloating in genetic programming. In: Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation - Volume 2, GECCO'99, pp. 1053–1060. Morgan Kaufmann Publishers Inc. (1999)

[10] Paris, G., Robilliard, D., Fonlupt, C.: Applying Boosting Techniques to Genetic Programming. Artificial Evolution 5th International Conference, Evolution Artificielle, EA 2001 **2310**, 267–278 (2001). DOI doi:10.1007/3-540-46033-0_22

References

[11] Souza, L.V.D., Pozo, A.T.R., Neto, A.C., Rosa, J.M.C.: Genetic Programming and Boosting Technique to Improve Time Series Forecasting (October) (2009)

[12] Potter, M.A., Jong, K.A.D.: A cooperative coevolutionary approach to function optimization. In: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature, PPSN III, pp. 249–257. Springer-Verlag, London, UK, UK (1994).

[13] B, J.P., Nguyen, S., B, M.Z., Johnston, M.: Evolving Ensembles of Dispatching Rules Using Genetic Programming for Job Shop Scheduling. EuroGP **1**, 92–104 (2015)

[14] Heger, J., Hildebrandt, T., Scholz-Reiter, B.: Dispatching rule selection with Gaussian processes. Central European Journal of Operations Research pp. 1–15 (2013).

[15] Mouelhi-Chibani, W., Pierreval, H.: Training a neural network to select dispatching rules in real time. Computers and Industrial Engineering **58**(2), 249 – 256 (2010).

[16] Nguyen, S., Zhang, M., Johnston, M., Tan, K.C.: Learning iterative dispatching rules for job shop scheduling with genetic programming. International Journal of Advanced Manufacturing Technology **67**(1-4), 85–100 (2013). DOI 10.1007/s00170-013-4756-9

[17] Joseph Y-T. Leung, Handbook of Scheduling: Algorithms, Models, and Performance Analysis