Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds.

SIMULATION-BASED OPTIMIZATION FOR SEMICONDUCTOR MANUFACTURING USING HYPER-HEURISTICS

Tobias Uhlig
Falk Stefan Pappert
Oliver Rose

Department of Computer Science
Universität der Bundeswehr München
Neubiberg, 85577, GERMANY

ABSTRACT

In semiconductor manufacturing we face many intricate scheduling problems. Simulation-based scheduling is a promising approach to deal with them. In conjunction with a metaheuristic we can solve many problem instances in a satisfactory manner. Nevertheless the quality of the results varies across the range of diverse challenges. Instead of performing extensive tests to determine the best metaheuristic and the optimal parameter setting for each case we propose the use of hyper-heuristics. A hyper-heuristic manages multiple metaheuristics to generate a solution for a broad field of applications. This paper will introduce two hyper-heuristics, one is an extended particle swarm algorithm the other is an integrated approach based on an evolutionary algorithm.

1 INTRODUCTION

A typical problem faced in semiconductor manufacturing is job scheduling. This challenge can be effectively tackled with simulation-based optimization. Solution candidates generated by a metaheuristic are simulated and evaluated according to a objective function. In an iterative process a suitable solution is attained. In this study we consider the issue of assigning jobs to parallel tools focusing on three basic problems:

- **Sequencing** deals with the problem of determining an processing order for jobs. This often directly influences the manufacturing process, considering constraints like sequence-dependent process and setup times. It furthermore has an high impact on objectives based on adherence to delivery dates.
- **Partitioning** of job sets considers the problem of assigning tasks to multiple available resources. A reasonable distribution of jobs to the tools is necessary to achieve load balancing.
- **Grouping** of jobs is concerned with arranging similar jobs for joint processing. The prime example for this problem is forming batches for batch-processing.

All of these problems are generally instances of the Knapsack problem which is proven to be NP-complete. Tackling real world issues is often even more difficult since they are a combination of the aforementioned basic problems. Furthermore one has to consider additional constraints during optimization, e.g., resource qualifications and priorities. Various metaheuristics exist which more or less successfully handle these problems. Choosing an appropriate one is a complex challenge by itself. Given a static problem, experience and experiments can guide the process of selecting an adequate metaheuristic. However considering a dynamic system with a shifting problem focus one can only settle for a solution which is suitable on average. It is desirable to automatically apply the best available metaheuristic for a given problem instance. A simplistic approach would employ more than one heuristics in parallel and use the best result generated. This approach is redundant and ignores possible synergies between multiple heuristics. Hence a more

978-1-4673-4781-5/12/$31.00 ©2012 IEEE

sophisticated approach like hyperheuristics is desirable. According to Burke et al. (2010) a ”hyper-heuristic is an automated methodology for selecting or generating heuristics to solve hard computational search problems.” It is a high-level approach which does not directly search for a solution, but aims to automate the design and adaptation of other heuristics. A hyper-heuristic tries to amplify the strengh of the employed heuristics and cover their weaknesses to raise the level of generality and solve complex problems (Burke et al. 2010). The following sections will introduce two different ideas to implement hyperheuristics.

2 META PARTICLE SWARM OPTIMIZATION

Particles Swarm optimization is inspired by foraging birds. Each particle represents a candidate solution in the search space (Kennedy and Eberhart 1995). They explore the search space relying on simple rules, i.e. move towards global optimum, move to the particles previously best found solution and random movement. We expand this concept by replacing the simple particles with complete metaheuristics. Each of them should behave like a particle in the particles Swarm optimization algorithm and accordingly implement the three basic movement rules. Random movement and incorporation of the best known local optimum are already inherent properties of most existent metaheuristics. Therefore we only need to add the exchange of knowledge with regard to the best known global solution. This results in metaheuristics exploring the search space independently while at the same time communicating with each other about promising solutions. Effectively infusing metaheuristics with results from other ones, generates a mutual benefit and therefore improves the overall performance. Furthermore this approach can easily be implemented in a decentralized and distributed way making it a perfect choice for modern parallel computer architectures and networks.

3 AN INTEGRATED HYPERHEURISTIC USING SELF-REPLICATING INDIVIDUALS

An evolutionary algorithm is a population based metaheuristic, using variation and selection to iteratively generate better solutions. Variation is used to derive new candidate-solutions from already existing ones relying typically on operations like recombination and mutation. Selection steers the evolutionary process by propagating better solutions and eliminating the bad ones. Consequently one can expect the quality of solutions to increase with each successive population. Each population consists of many individuals representing possible solutions to the problem. In a conventional evolutionary algorithm these individuals are mere containers of data encoding a candidate solution. Today’s common approach is to apply external methods to individuals in order to generate offspring. Our approach is based on the idea of self-replicating individuals, moving all of the reproduction-logic into the individuals. This is much closer to natural evolution where each individual is indeed an acting agent. For this work we utilize the capabilities of this approach to generate an integrated hyper-heuristic. Confining the reproduction logic into the individuals enables us to use different logic for various individuals (Uhlig and Rose 2011). We can use inspiration from existing metaheuristics to generate different kinds of individuals. These kinds are like species or races in natural evolution and in accordance with the customary evolutionary algorithms they compete with each other. Selective pressure will eventually promote the ones which were inspired by the most adequate metaheuristics. On the other hand the use of recombination during the replication process facilitates the merging of beneficiary traits across different kinds of individuals.

REFERENCES

Burke, E. K., M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, and J. R. Woodward. 2010. “A Classification of Hyper-heuristic Approaches”. 449–468. Springer.

Kennedy, J., and R. Eberhart. 1995, November. “Particle swarm optimization”. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, Volume 4, 1942–1948 vol.4: IEEE.

Uhlig, T., and O. Rose. 2011. “A Multi Species Evolutionary Algorithm for Tool Group Scheduling in Semiconductor Manufacturing”. In *Proceedings of the 5th Multidisciplinary International Conference on Scheduling: Theory and Applications*, edited by J. Fowler, G. Kendall, and B. McCollum, 459–468.