Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

SIMULATION-BASED MULTI-OBJECTIVE OPTIMIZATION OF A REAL-WORLD SCHEDULING PROBLEM

Anna Persson
Henrik Grimm
Amos Ng
Thomas Lezama

Centre for Intelligent Automation
University of Skövde
Box 408, 541 28, SWEDEN

Jonas Ekberg
Stephan Falk
Peter Stablum

Swedish Postal Services, Posten AB
105 00 Stockholm, SWEDEN

ABSTRACT

This paper presents a successful application of simulation-based multi-objective optimization of a complex real-world scheduling problem. Concepts of the implemented simulation-based optimization architecture are described, as well as how different components of the architecture are implemented. Multiple objectives are handled in the optimization process by considering the decision makers' preferences using both prior and posterior articulations. The efficiency of the optimization process is enhanced by performing culling of solutions before using the simulation model, avoiding unpromising solutions to be unnecessarily processed by the computationally expensive simulation.

1 INTRODUCTION

Posten AB is the Swedish postal services, entirely owned by the Swedish Government. Core business comprises distribution of messages and logistics, and Posten is one of the largest actors in these areas in the Nordic region. As the Nordic postal market is fully deregulated, mail business is a highly competitive market. Facing national and international actors operating in the same business areas puts high demands on efficient mail operations, and additional pressure arises from legal directives that Posten is obligated to follow, specifying that mail operations must be fast, reliable, and cost-efficient.

Every day, Posten receives over 22 million pieces of mail and before distribution to recipients, all mail is sorted in delivery order. The sorting is highly automated, carried out using a set of machines that scan mail and determine destination address using sophisticated optical character reading. For sorting to be efficient, mail are divided into batches of 10,000-35,000 pieces according to destination region and these batches are sorted in parallel on different machines.

An operation schedule defines for each mail batch what machine to use for sorting and on which time to start the sorting. Since mail batches, as well as machines, have different characteristics, consequences of different schedules – such as time consumption and money expenses – vary a great deal. In Figure 1, a simplified operation schedule is presented.

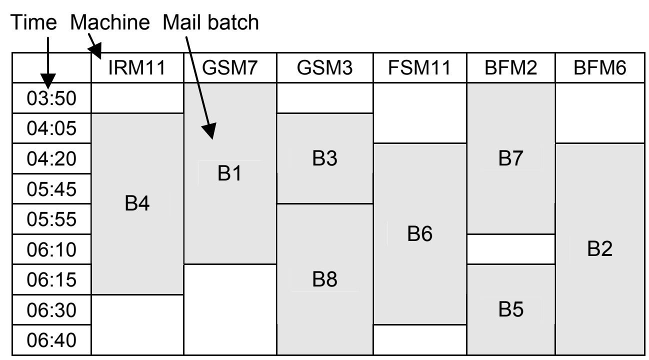

Figure 1: Simplified Mail Operation Schedule

As a convenient way to observe mail sorting operations and consequences of different operation schedules, Posten has developed a discrete-event simulation model of the mail sorting process. Given a schedule as the input, the model can report its performance in less than a minute. The simulation model, developed using Arena simulation software (www.arenasimulation.com), allows testing and analysis of different schedules without disrupting the real system. The simulation model is exclusively an evaluation tool and supports neither generation nor optimization of schedules.

The established method for creating schedules has so far been a manual procedure based on trial and error. As there is no guidance on how to change input parameters between iterations, this approach is very time-consuming and requires many iterations and extensive effort by an expert for finding a satisfactory schedule. Furthermore, it does not guarantee that a valid schedule is found, but leaves the validation entirely in the hands of the expert, who is required to consider and carefully control all possible constraints. As there are multiple conflicting objectives to consider when creating a schedule, manual optimization is practically impossible – especially since explicit heuristics for finding a good schedule is missing.

This paper presents how the scheduling process and the resulting schedules have been improved by implementing an automatic simulation-based system that supports the generation of optimized operation schedules while considering multiple objectives simultaneously.

The rest of the paper is organized as follows. The next section presents related work in the area of simulation-based multi-objective optimization of operation schedules. In Section 3 the multi-objective problem considered is described, followed by a description of the optimization approach used for solving it in Sections 4 and 5. Section 6 describes the simulation-based optimization system implemented and its components. Conclusions of the paper and a description of planned future work are presented in Section 7.

2 RELATED WORK

Multi-objective optimization is an active research area in the field of optimization methodologies, particularly using Evolutionary Algorithms (Deb 2001). Nevertheless, the literature reports relatively few attempts in the area of simulation-based multi-objective optimization. Exceptions can be found in (Eskandari et al. 2005) and (Baesler and Sepúlveda 2001). Only a small fraction of the papers in the literature, however, consider operation scheduling problems.

Almeida et al. (2001) use a simulation-based approach for multi-objective optimization of operation schedules in a petroleum refinery. Their proposed method is based on a genetic algorithm combined with a multi-objective energy minimizing method. Using the method the authors succeeded in finding a schedule for a real-world refinery production problem with three objectives; maximization of diesel production, maximization of jet fuel production, and minimization of costs.

Allaoui and Artiba (2004) propose a method based on a combination of simulated annealing and dispatch rules for simulation-based multi-objective optimization of flow shop schedules. In this method, both stochastic and deterministic unavailability of machines are considered in the optimization strategy. The authors applied their proposed method for solving a NP-hard scheduling problem with the aim of optimizing work in progress, job tardiness, and utilization of resources.

Gupta and Sivakumar (2002) present a simulation-based multi-objective optimization method based on compromise programming for operation scheduling in semiconductor manufacturing. The proposed method was applied for finding a Pareto optimal solution to a NP-hard problem of scheduling a number of independent jobs on a single machine. The objectives considered in the optimization strategy were average cycle time, average tardiness, and machine utilization. A number of theoretical job-shop experiments was successfully carried out using the proposed method.

3 PROBLEM DESCRIPTION

This sections presents the operation scheduling problem considered in this paper. In general terms, the problem of finding an optimal mail operation schedule can be described as follows. There are J non-identical, independent jobs (i.e. mail batches) to be assigned to M non-identical, independent parallel machines. No explicit job priorities exist. The machines are available at different times and each machine can process only one job at a time. Job preemption is not allowed; once a job starts on a machine it must be processed to completion. A job is ready for processing at its release time and must be completed before its deadline. The processing time of a job depends on the machine it is assigned to and varies between different machines. Machine capacity must be respected and a job cannot be assigned to a machine in which the capacity is below the processing requirements of the job.

Jobs should be assigned to machines in a way that is optimal according to a balanced relationship between:

- Total money expenses: machine usage is associated with money expenses for machine wear, electricity, etc. and the total expenses should be as small as possible.
- Total slack time of jobs: the time between the completion of a job (mail batch) and its deadline should be as long as possible, allowing wider margins for the distribution of mails to recipients.
- Load balancing of machines: jobs should be distributed to machines in a way that promotes even utilization..

Most scheduling problems belong to a class of problems that is called NP-complete (Azzaro-Pantel et al. 1998), which means that the time required for computing an optimal solution increases exponentially with the size of the problem. This property also applies to the problem considered in this paper, as the simplified problem of scheduling a set of J uninterruptible jobs so that the jobs are completed before their deadlines on machines that are capable to process only one job at a time, is NP-complete in the ordinary sense (Cormen et al. 2001). NP-complete problems are computationally expensive since guaranteeing an optimal solution requires an exhaustive search in which all possible solutions have to be tried and evaluated. Since such an exhaustive search takes unreasonable computing time for most complex scheduling problems, a common and acceptable practice is to sacrifice optimality for efficiency by heuristically guiding the search and evaluating only a fraction of all configurations (Arnaout and Rabadi 2005).

4 MULTI-OBJECTIVE OPTIMIZATION

In general multi-objective optimization problems, there exist no single best solution with respect to all objectives as improving performance on one objective deteriorate performance of one or more other objectives (Srinivas and Deb 1995). This is also the case for the multi-objective optimization problem considered in this paper.

As it is not possible to obtain solutions which maximize performance of all objectives at the same time, the optimal solutions to the problem are considered to be the Pareto optimal set. The Pareto optimal solutions are the set of solutions strictly superior to the other solutions considering all objectives but possibly inferior to other solutions considering one or a subset of the objectives (Srinivas and Deb 1995). Any of the Pareto optimal solutions is an acceptable solution, since none of them is absolutely superior to any others (Srinivas and Deb 1995).

4.1 Two-Stage Articulation of Preferences

Various methods for simulation-based multi-objective optimization can be categorized according to the timing of when the articulation of the required preference information occurs relative to the optimization (Evans et al. 1991). This timing can be:

- Before the optimization (prior articulation of preferences)
- During the optimization (progressive articulation of preferences)
- After the optimization (posterior articulation of preferences)

None of these alternative approaches is generally better than another for solving multi-objective problems, but they all have various strengths and weaknesses (Evans et al. 1991). In this paper, we use a combination of prior and posterior articulation of preferences; benefiting the strengths of both these approaches. We do not make use of progressive articulation of preferences as it is considered too time-consuming for the decision maker to be involved during the entire optimization process. While many multi-objective optimization methods are based on either prior or progressive articulation of preferences, only a few attempts have been made based on posterior articulation of preferences (Medaglia et al. 2004).

Prior to the optimization, the decision maker is asked to express tradeoff preferences regarding the various objectives by assigning a weight value to each objective specifying its relative importance – a higher value means that the objective is considered more important. From a total amount of 100% the decision maker allots each objective a percentage value and the total weighting assigned must sum up to 100%, as shown in the example in Table 1. This tradeoff information is used to determine the direction of the optimization strategy, making the optimization process more efficient.

Table 1: Example of Objectives Weighting

Objective	Weight
Minimize total money expenses	50
Maximize slack time of jobs	30
Maximize machine load balancing	20

Posterior to the optimization, all identified Pareto optimal solutions are presented to the decision maker with information of their achievement level of the various objectives. The decision maker may then choose the most desirable one from the solution set using some other higher-level information based on his/her domain knowledge.

4.2 Integrating Multiple Objectives

The various objectives are weighted by the decision maker prior to the optimization and aggregated into a single objective through a weighted sum function (Weigert et al. 2000):

$$v = \sum_{i=1}^{n} w_i u_i \text{ , where } \sum_{i=1}^{n} w_i = 1, w_i \geq 0 \text{ ,} \tag{1}$$

and where u_i is the subutility produced through objective i, w_i is the relative importance of objective i, and n is the number of objectives. The goal of the optimization strategy is to maximize v considering all problem constraints.

4.2.1 Normalization of Objectives

The various objectives considered are all represented using different measurement units. To allow a fair comparison between performance of the different objectives, all objective measurements are normalized to values between 0 and 1:

$$u_i = \frac{o_i - worst_i}{best_i - worst_i}$$

where u_i is the utility of objective o_i, o_i^{out} is the measured output value of o_i, $worst_i$ is the worst possible value of o_i, $best_i$ is the best possible value of o_i and an optimal value of u_i is 1.

5 OPTIMIZATION ALGORITHM

The multi-objective optimization strategy is based on a Genetic Algorithm (GA). GAs are population based search algorithms inspired by theories from natural evolution. The basic idea behind these algorithms is that a population of individuals represents possible solutions of a given problem. Through recombination of solutions, offspring are created, forming a new generation of the population. Some of the solutions are better suited for the problem and these are given more opportunities to reproduce and pass their desirable behavior to the next generation, similar to natural selection. New generations of the population are evolved until a sufficiently good solution is found.

GAs have been proven to be very flexible and reliable in searching for global solutions (Baesler and Sepulveda 2000) and also capable of solving complex scheduling problems (Azzaro-Pantel et al. 1998). Their characteristics making them suitable for solving multi-objective simulation-based problems (Eskandri et al. 2005) and they can easily be coupled with any discrete-event simulation models, in contrast with some other heuristic methods which are more suitable only to certain problems (Azzaro-Pantel et al. 1998).

The rest of this section describes the GA implemented for solving the multi-objective optimization problem considered in this paper. The implementation was based on GAlib (`http://lancet.mit.edu/ga/`), which is a C++ library of GA components.

5.1.1 Representation of Solutions

The GA encodes possible solutions as genomes and each genome instance represents a single solution to the problem – in this case an operation schedule. In many applications, the efficiency of GAs is determined mainly on how the domain problem is encoded in the genome and the representation has therefore been considered carefully in this study. For this problem, the set of jobs is considered to be $J=\{j_1, j_2, \ldots, j_k\}$ and the set of machines $M=\{m_1, m_2, \ldots, m_n\}$. A genome consists of n lists of variable length, $\left(\left(j_{1,1}, j_{1,2}, \ldots, j_{1,i_1}\right), \ldots, \left(j_{n,1}, j_{n,2}, \ldots, j_{n,i_n}\right)\right)$, where each list represents scheduling information for a specific machine. Each list entry represents a job scheduled on the machine (e.g., jobs $j_{1,1}, j_{1,2}, \ldots, j_{1,i_1}$ are scheduled on machine 1). The genome is a permutation of all jobs, i.e., each job is present in one and only one of the lists.

When generating a schedule from a genome, each job list is sorted first by starting time and subsequently by deadline. The assumption behind this approach is that a job with an earlier starting time has an earlier deadline and thus there is no reason to schedule an early job after a late job. When a list has been sorted, jobs are scheduled on the machine in the sorted order. Figure 3 shows an example of a simplified genome with and fourteen jobs scheduled on four machines.

m_1	j_9	j_1			
m_2	j_{14}	j_{13}	j_3	j_5	
m_3	j_7	j_2	j_4	j_{12}	j_{11}
m_4	j_6	j_8	j_{10}		

Figure 3: Example of Genome

An advantage of this representation is that the genetic material used to represent a solution (i.e. an operation schedule) is kept small, reducing the size of the search space and thus improving performance of the algorithm. A drawback, on the other hand, is that it may represent infeasible solutions – however this is handled by the evaluation function, giving only partial credit to infeasible solutions.

5.1.2 Genetic Operators

A first population of 50 candidate solutions is randomly created. In the initialization procedure, a heuristic function is used to make sure that jobs are only scheduled on machines with enough capacity. During each successive generation of the GA, a proportion of the existing population is selected to breed a new generation. Individual solutions are chosen for mating through roulette wheel selection, in which the probability for selection is proportional to the fitness of the solution. Thus, solutions with higher fitness values are more likely to be selected, but a small number of solutions with less fitness values have some probability of being selected as well in order to keep a large diversity of the population.

From the pool of selected solutions, two solutions are chosen as parents and through mating two new solutions are formed according the procedure outlined in Figure 4. This process, called crossover, takes place with a probability of 0.9.

```
foreach j in J do
  machine1 ← WhichMachine(parent1, j)
  machine2 ← WhichMachine(parent2, j)
  with probability 0.1 do
    Swap(machine1, machine2)
  end
  AddToList(child1[machine1], j)
  AddToList(child2[machine2], j)
end
```

Figure 4: Crossover Function

To maintain the genetic diversity from one generation to the next, some of the offspring solutions are mutated. In the mutation procedure, all jobs are iterated through and a job is moved to a random list with a probability of 0.1. Similar to the initialization procedure, a heuristic is used when mutating solutions to make sure that jobs are only scheduled on machines with enough capacity.

5.1.3 Fitness Function

A fitness function quantifies the quality of solutions by assigning a fitness value to each of them corresponding to their performance. The fitness value assigned is based on a combination of two properties of a solution:

- Objectives: Credit is given based on the achievement of the objectives; the higher the level of achievement, the higher the credits.
- Delay: If the processing of a job is completed after its deadline a penalty is given.

Based on these two properties, the formula to calculate the fitness of solution is:

$$f = v - d\,w_d \qquad (2)$$

where v is the result of the weighted sum function in Equation (1), d is the number of delays, and w_d is the weight for property d.

5.1.4 Evolutionary Process

The overall GA evolution process works as describe below. A flow diagram of the process is also presented in Figure 4:

1. A first population of candidate solutions (i.e. operation schedules) is created. These initial solutions are randomly generated in order to enable a wide range of solutions. To achieve faster improvement of the algorithm it is also possible to insert user-defined solutions known to have good performance in the initial population.
2. Performance of solutions are evaluated and a fitness value is assigned to each of them using the formula described in Equation (2).
3. The solutions with highest fitness are selected and by applying genetic operators, offspring are created from these solutions, forming the new, improved generation of the population.
4. Some of the solutions in the new generation are arbitrary mutated to maintain the genetic diversity.
5. If a the stopping criterion is met, the search is terminated and the set of Pareto optimal solutions are returned – otherwise the process is repeated from step 2.

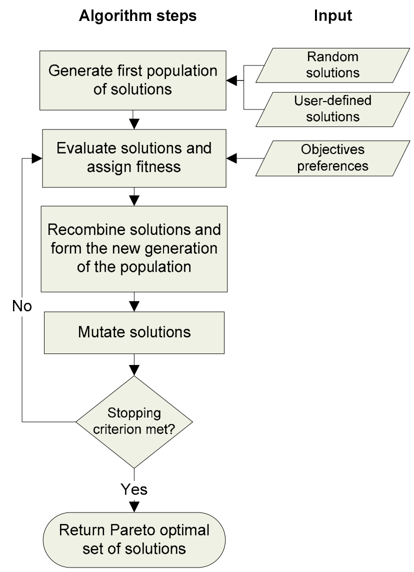

Figure 5: Optimization Strategy

6 DESIGN AND IMPLEMENTATION

This section describes the realization of a simulation-based system that supports the automatic generation of optimized operation schedules.

6.1 Architectural Design

The suggested solution is based on the architecture shown in Figure 6. The decision maker specifies optimization objective preferences and inputs these to a client application, which initiates the optimization process and sends the objectives to an optimization component. A candidate solution to the problem is automatically generated by the optimization component and sent to the evaluation component. The evaluation component quantifies the performance of the suggested solution and notifies the optimization component of the results. Based on this feedback, the optimization component generates an improved solution and sends this new solution to the evaluation component. The generate-and-evaluate process is then repeated until the stopping criteria is met and when this happens the resulting set of Pareto optimal solutions is sent back to the client component, where the results are presented to the decision maker.

The architecture is based on the principle of encapsulation with low couplings between components; components only know about each others' interfaces (i.e. input and output) with minimal knowledge of their internal details.

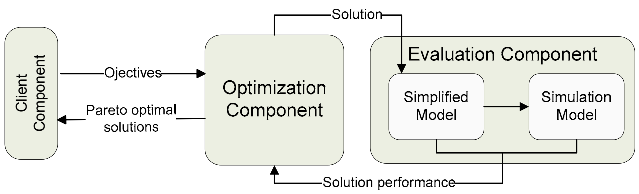

Figure 6: Architectural Design

The low coupling between components allows for a flexible implementation where components can be changed without influencing each other, assuming that the interface is unchanged. It also fits well into a distributed and parallel computing platform where different components can be run on separate computers.

The architecture is easy to use, as the user only interacts with a simple interface and does not need to have any knowledge of the optimization strategy. The user is replaced in the problem-solving process by the optimization component and besides specification of optimization objective preferences, no manual effort is required.

6.2 Implementation

This section describes how the system presented in the previous section has been implemented.

6.2.1 Client Component

The client component was implemented using Excel and Visual Basic for Applications (VBA), as Excel was the already existing interface for the Arena simulation model. When the client is started, it prompts the user to input details about optimization objective preferences. The user then starts the optimization process by clicking a button.

When the optimization process is initiated, a VBA script generates a text file specifying optimization objectives tradeoffs, sends this file in a call to the optimization component, and waits for the optimization component to send back the set of Pareto optimal schedules. The resulting schedules are presented to the user in the Excel interface together with summary results and statistics.

6.2.2 Evaluation Component

The evaluation component consists of two subcomponents; a Simplified Model and an Arena simulation model. The idea of introducing a Simplified Model, and not only using a simulation model as in ordinary simulation optimization approaches, is to reduce the overall time consumption needed for evaluation of solutions. The simulation model is the main bottleneck in the process and avoiding unnecessary use of it will enhance the system efficiency. The Simplified Model performs a rough estimation of solutions and does not consider stochastic events in the system.

Solutions sent to the evaluation component are first processed by the Simplified Model, which performs culling of unpromising solutions and acts as filter to the simulation subcomponent. The Simplified Model approximates the time consumption of each job in a schedule and estimates if all deadlines are met, i.e. if a schedule is valid. If a solution is considered as invalid by the Simplified Model, it is not sent to the time-consuming simulation for further evaluation, but feedback is returned to the optimization component immediately. As the Simplified Model is an approximation of the simulation there is an inherent risk that solutions are misclassified. A false-positive classification (i.e. when an invalid solution is classified as valid) cause no harm on the optimization results but only add some extra time to the process. A false-negative classification (i.e. when a valid solution is classified as invalid), on the other hand, has the consequence that promising solutions may not proceed in the optimization process. To reduce the number of false-negative classifications, the classification procedure of the Simplified Model is made optimistic.

Solutions considered as valid by the Simplified Model are sent to the Arena simulation model for detailed evaluation of processing times, money expenses, and other properties depending on complex interrelationships between different parts of the system, often influenced by stochastic events. Before the Arena simulation model was inserted in the system it was carefully verified and validated, since the correctness of the simulation model is of critical importance for the optimization results to be useful in reality.

6.2.3 Optimization Component

The optimization component, based on the optimization strategy described in Section 5, uses the evaluation component for performance quantification of solutions. The optimization component is not aware of in what way solutions are evaluated, but only receives performance quantifications.

6.3 Results

The implemented system was tested using real-world scenarios and so far the results look very promising. Evaluation of results from the tests show that the system is successful in finding good schedules according to optimization objectives preferences specified by the decision maker. Domain experts have compared the generated schedules with their own manually created schedules and see a great potential in the system.

Compared to the manual approach of creating schedules, the system implemented has considerable advantages. Since no manual control or intervention in the optimization process is necessary, a lot of time and effort are saved for the expert responsible for creating schedules. The system also makes it easy to obtain schedules with certain focuses, such as for example low money expenses, as the decision makers' preferences are considered in the optimization process. It is worth to notice, however, that from the results it is not clear how well the optimized solutions compare with the achievable values of the objectives, as the ideal values are not known.

By performing culling of solutions before the time-consuming simulation takes place, a reduced time consumption for the total evaluation process can be achieved. As shown in the chart presented in Figure 8, the number of unpromising solutions tend to be large, especially in the beginning of the optimization process (results are an average of ten runs). An explanation of this is that the GA starts from a set of random solutions and performs a broad search over the whole search space. Note that not all individuals in the population are evaluated for each generation because some solutions remain unchanged from one generation the next.

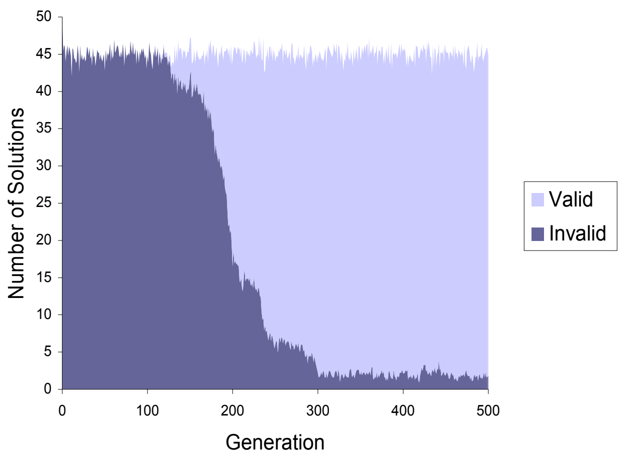

Figure 8: Number of Valid and Invalid solutions as Classified by the Simplified Model

7 CONCLUSIONS AND FUTURE WORK

This paper presents a successful application of simulation-based multi-objective optimization of a complex real-world operation scheduling problem. A two-stage articulation of the decision makers optimization objective preferences was used in the multi-objective approach. Expressing preferences prior to the optimization enables the direction of the optimization strategy to be influenced, making the search process more efficient. Presenting the complete set of the resulting Pareto optimal solutions posterior to the optimization enables the decision maker to choose the preferred one and hence results in a final solution that is desirable from the decision maker's perspective. As there may be quite many Pareto optimal solutions, it is important to present the set of solutions in a way that aids the decision maker in the task of analyzing all solutions to find the best one. In Persson et al. (2006) we describe how this support can be provided and present ideas of a graphical user interface for analysis of solutions.

The overall process was made more efficient by performing a rough estimation of solutions before evaluating them using the time-consuming simulation. Potential is seen in this approach of avoiding unpromising solutions to be unnecessarily evaluated. However, the culling of solutions requires further studies, which are included in planned future work.

To improve optimization results further, future work also includes studying how domain expert knowledge can be captured and incorporated in the optimization process. A human expert may have extensive knowledge valuable for the optimization, and incorporating this knowledge in the optimization strategy may be a way to obtain faster and more accurate optimization results.

ACKNOWLEDGEMENT

This work is done within the OPTIMIST project which is partially financed by the Knowledge Foundation (KK Stiftelsen), Sweden. It uses GAlib developed by Matthew Wall at the Massachusetts Institute of Technology.

REFERENCES

Allaoui, H., and A. Artiba. 2004. Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints. *Computers & Industrial Engineering* 47: 431-450.

Almeida, M. R., S. Hamacher, M. A. C. Pacheco, and M. B. R. Velasco. 2001. Applying Genetic Algorithms to the Production Scheduling of a Petroleum Refinery. In *MIC'2001 - 4th Metaheuristics International Conference*, 773-777.

Arnaout, J-P. M., and G. Rabadi. 2005. Minimizing the Total Weighted Completion Time on Unrelated Parallel Machines with Stochastic Times. In *Proceedings of the 2005 Winter Simulation Conference*. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Azzaro-Pantel, C., L. Bernal-Haro, P. Baudet, S. Domenech, and L. Pibouleau. 1998. A two-stage methodology for short-term batch plant scheduling: discrete-event simulation and genetic algorithm. *Journal of Computers and Chemical Engineering* 22(10): 1461-1481.

Baesler, F. F., and J. A. Sepúlveda. 2001. Multi-Objective Simulation Optimization for a Cancer Treatment Center. In *Proceedings of the 2005 Winter Simulation Conference*. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Cormen, T. H., C. E. Leiserson, R. L. Rivest, and C. Stein. 2001. *Introduction to Algorithms*. 2nd edition. USA: MIT Press.

Deb, K. 2001. *Multi-objective Optimization Using Evolutionary Algorithms*. Chichester: John Wiley & Sons.

Eskandari, H., L. Rabelo, and M. Mollaghasemi. 2005. Multiobjective Simulation Optimization Using an Enhanced Genetic Algorithm. In *Proceedings of the 2005 Winter Simulation Conference*. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Evan, G., Stuckman, M. and Mollaghasemi, M. 1991. Multiple response simulation optimization. In *Proceedings of the 1991 Winter Simulation Conference*. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Gupta, A. K., and A. I. Sivakumar. 2002. Simulation based Multiobjective Schedule Optimization in Semiconductor Manufacturing. In *Proceedings of the 2002 Winter Simulation Conference*. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Medaglia, A. L., S. B. Graves, and J. L. Ringuest. 2004. Multiobjective evolutionary approach for linearly constrained project selection under uncertainty. Technical Report No. COPA 2004-003, Department of Industrial Engineering, University of Los Andes, Colombia.

Persson, A., H. Grimm, and A. Ng. 2006. On-line Instrumentation in Simulation-based Optimization. In *Proceedings of the Winter Simulation Conference 2006*. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Srinivas, N., and K. Deb. 1995. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation* 2(3): 221–248.

Weigert, G., S. Werner, D. Hampel, H. Heinrich and W. Sauer. 2000. Multi Objective Decision Making – Solutions for the Optimization of Manufacturing Processes. In *Proceedings of the 10th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2000)*, 487 - 496.

AUTHOR BIOGRAPHIES

ANNA PERSSON is a Ph.D. candidate at University of Skövde, Sweden and De Montfort University, U.K. She holds a Master's degree in Computer Science from University of Skövde. Her research interests include artificial intelligence, simulation-based optimization, and efficiency enhancement techniques for simulation-based optimization. Her e-mail address is <anna.persson@his.se>.

HENRIK GRIMM is a Systems Developer at the University of Skövde, Sweden. He received his BSc and MSc degrees in Computer Science from University of Skövde. His research interests include computer simulation, artificial intelligence, and distributed systems. His e-mail address is <henrik.grimm@his.se>.

AMOS H. C. NG is a Senior Lecturer at the University of Skövde, Sweden. He holds a B.Eng. degree and a M.Phil. degree, both in Manufacturing Engineering from the City University of Hong Kong and a Ph.D. degree in Computing Sciences and Engineering from De Montfort University, Leicester, U.K. He is a member of the IEE and a Chartered Engineer in the U.K. His research interests include agent-based machine control systems, virtual engineering for manufacturing machinery and machine systems as well as simulation-based optimization. His e-mail address is <amos.ng@his.se>.

THOMAS LEZAMA is a Lecturer at the University of Skövde. He holds a B.Sc. in Automation from University of Skövde, Sweden and a M.Sc. in Computer Integrated Manufacturing from Loughborough University, Loughborough, U.K. 2005 he received a Ph.D. degree in Computing Science and Engineering from De Montfort University, Leicester, U.K. His e-mail address is <thomas.karlsson@his.se>.

JONAS EKBERG works within the Logistics Development Department at Posten AB. He has 7 years of experience from process development with simulation- and optimization tools, including development, project management, analysis and training. He holds an M.Sc. in Systems Engineering from the Royal Institute of Technology in Stockholm. His e-mail address is <jonas.ekberg@posten.se>.

STEPHAN FALK is Project Manager at Posten AB where he is responsible for among others simulation tools for the production. His professional background concerns primarily operational development and business development responsibilities within Ericsson and Emerson. He holds a BScEE and a MBA degree. His e-mail address is <stephan.falk@posten.se>.

PETER STABLUM works within the Logistics Development Department in Posten AB. The last couple of years, supply chain simulation and optimization have been his main tasks. He holds an M.Sc. in Industrial Engineering and Management from Luleå University of Technology. His e-mail address is <peter.stablum@posten.se>.