Environment. Technology. Resources, Rezekne, Latvia
Proceedings of the 10th International Scientific and Practical Conference. Volume III, 175-179

Job Shop Production Planning under Uncertainty: A Monte Carlo Rollout Approach

Axel Simroth[1], Denise Holfeld[1], Renè Brunsch[2]
[1]Fraunhofer Institute for Transportation and Infrastructure Systems IVI
[2]apromace data systems GmbH

Abstract. **The Monte Carlo Rollout method (MCR) is a novel approach to solve combinatorial optimization problems with uncertainties approximatively. It combines ideas from Rollout algorithms for combinatorial optimization and the Monte Carlo Tree Search in game theory. In this paper the results of an investigation of applying the MCR to a Scheduling Problem are shown. The quality of the MCR method depends on the model parameters, search depth and search width, which are strong linked to process parameters. These dependencies are analyzed by different simulations. The paper also deals with the question whether the Lookahead Pathology occurs.**

Keywords: **optimization under uncertainty, robust optimization, Monte Carlo Rollout method, combinatorial problems, Scheduling Problem.**

I INTRODUCTION

Many challenges have to be solved, when applying optimization methods to industrial problems. Possible inaccuracies in information are one of the major challenges. In general, information is not completely known a priori, like it is assumed in mathematical optimization theory. Often information changes over time or can only be estimated. Even solutions must be provided in real time. Both aspects are relevant for many industrial applications like the Job Scheduling Problems.

The Monte Carlo Rollout method (MCR) has been used successfully in various industrial applications to solve combinatorial problems approximately. Here, the problem is considered, to assign different jobs to different machines as well as the order of the production. Machine breakdowns can occur, which lead to an interruption of the production ([1]). There are several possibilities for the distribution of jobs to the machines. Which distribution should be chosen to minimize the delay? The knowledge on uncertainties should be included into the planning. Therefore a stochastic model is necessary to model uncertainties, with the aim to assign probabilities to different possible future developments.

II PROPOSED APPROACH

In order to make decisions in real time, a heuristic *H* is used to determine a solution based on current information in many cases. The idea is to improve the solution by a Monte Carlo Rollout (MCR) approach as meta-heuristic. In this way, the heuristic is combined with a stochastic model for simulating future developments. This leads to better and more robust solutions for optimization problems under uncertainties. The MCR approach combines ideas from Rollout algorithms for combinatorial optimization and the Monte Carlo Tree Search in game theory. Therefore, we want to introduce all three methods.

Rollout algorithms ([2], [3]) can be used for optimization problems that have a sequential structure, i.e. which can be solved by making a sequence of consecutive decision steps with a limited number of alternative decisions in each step. With an infinite number of alternatives a finite selection must be made or an abort criterion must be selected. These optimization problems can be represented as a search tree. Each node is regarded as a decision point and each leave of the tree represents a solution. By means of the Rollout method, each alternative decision is evaluated in order to choose the best alternative. The Rollout algorithm iteratively explores all different alternatives in the current decision step. It uses a so-called base heuristic for making decisions in the steps following the current decision. The base heuristic usually is a fast, rather simple but solid heuristic for the problem at hand, that solves the problem in a sequential manner. With the help of this base heuristic the Rollout algorithm gets an evaluation of the alternative at a leaf of the decision tree, namely at that leaf that would be reached if the base heuristic would be applied after choosing the alternative considered.

ISSN 1691-5402

DOI: http://dx.doi.org/10.17770/etr2015vol3.617

After evaluating all alternatives in the current step, the one that leads to the best results is chosen. In this way only a small part of the tree is examined and the heuristic guides into the most promising part of the tree. If the problem contains *N* decisions, after *(N-1)* applications of the Rollout algorithm a path from root to leaf will be obtained. This path represents the approximated solution of the problem. In Fig.1 the process of a Rollout evaluation for *n* alternatives is shown schematically. For sequentially consistent heuristics, it can be shown, that the Rollout approach leads to no worse results than the pure application of sequentially consistent heuristic.

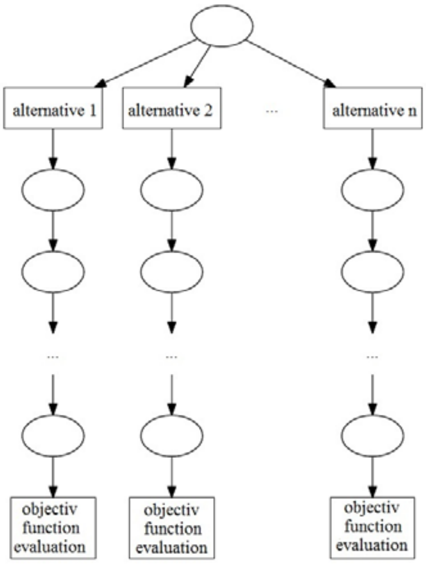
alternative 1
alternative 2
...
alternative n
...
objectiv function evaluation
objectiv function evaluation
objectiv function evaluation

Fig. 1. Schematic representation of a Rollout algorithm.

Monte Carlo Tree Search (MCTS) ([4], [5], [6]) is used for problems where no good heuristic was found to evaluate a decision. Instead of using a noisy and possible misleading heuristic evaluation, the alternative decisions are evaluated by means of random games. Often there is game-specific information about the quality of moves which can be used to weight the possible moves. The MCTS algorithm includes four steps which are shown in Fig.2. The steps are Selection, Expansion, Simulation and Back-propagation and they are repeated several times to generate the MCTS tree.

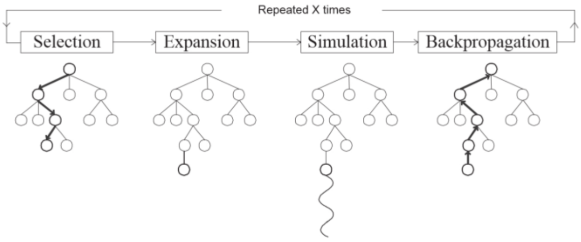

Fig. 2. Schematic representation of a MCTS algorithm.

In the first step a node which is not been included into the MCTS tree yet is selected. This selected node is added to the MCTS tree in step 2, which is the starting point of step 3. To avoid an inexact heuristic evaluation the evaluation of a node is carried out by simulation, also called play-outs. Therefore, all further decisions are taken at random. Such random game ends in a leaf which can be rated clearly. In the last step the MCTS tree is updated with the information from the simulation, e.g. by using the Mimimax algorithm, like [7]. These 4 steps are repeated up to a maximum. The maximum can be the number of nodes in the MCTS tree or the runtime of the algorithm.

The Monte Carlo Rollout method combines both approaches to handle sequential optimization problems that are afflicted with uncertainties. The evaluation of an alternative decision by solving the problem further with a simple and fast base heuristic is adapted from the Rollout approach. The uncertainties are covered through the random selection of future situations, by means of a random player as in the Monte Carlo Tree Search. So, the optimization problem with uncertainties is modeled as a two-player game. The first player is the decision maker, which decides on the base of a simple heuristic. The second player is the random player, which creates new future situations by random. The game where both players move consecutively is called MCR. With a set of different MCRs, an alternative is tested by evaluating a set of random future scenarios. So, the long-term behavior and robustness against uncertainties of this alternative could be analyzed. The MCR method is shown schematically in Fig.3.

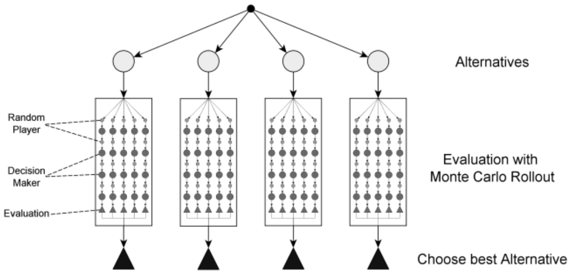

Fig. 3. Schematic representation of a MCR algorithm.

III APPLICATION

Considering the application of the Monte Carlo Rollout method to a Scheduling Problem, more precisely to a Stochastic Dynamic $(m \times l)$ Job Shop Problem with due date. Here, several plans are available, each with a static heuristic evaluation, which results from an objective function. Over time, however, machine breakdowns occur, which lead to delays. The MCR method is used for decision support to choose a robust plan against random future failures.

Considering jobs $J_1, \dots, J_n$ with given processing duration $d(J_i) \geq 0$. Each job J_i consists of sub jobs $J_{ij}, j = 1, \dots, m$ with $0 \leq d(J_{ij}) \leq d(J_i)$ and $\sum_{j=1}^{m} d\left(J_{ij}\right) = d(J_i)$. In the $(m \times l)$ Job Shop Problem

n jobs on m different types of machines are to be processed. Each job consists of m sub jobs with true positive processing durations. Each sub job is processed on one machine type. Each of the m types of machines has capacity l. Each job J_i has a fixed predetermined due date $\mathrm{F(J_i)}$. This is transferred to its sub jobs $F(J_{ij})$, according to the follow-up relations. The aim of the scheduling is to determine the execution time for each sub-job of all jobs, taking into account capacity constraints. A scheduling plan is feasible, if all jobs are completed by the due date and all capacity limits are observed.

To minimize the total delay is an appropriate objective function. So, every feasible and complete plan is evaluated by this objective function and the plans are comparable. This problem can be seen as a sequential decision process and is represented by a decision tree. The root of the tree is an empty plan. Each edge adds a sub job. Hence, each knot is a sub plan and each leave is a plan. If a job is added to the plan, the execution time and the assignment to a machine for each sub-job have to be determined. Decisions regarding the execution time of a job are selected in order to minimize the objective function, taking into account follow-up conditions and capacity constraints.

Heuristics follow only a single path of the decision tree and result in sub optimal planning. Possible heuristics are:

H_1: Starting each sub job as early as possible and choose the earliest predetermined due date of all unassigned jobs $I \subseteq \{1, \ldots, n\}$:

$$\mathrm{J}_k \text{ with } k = \arg\min_{i \in I} F(J_i)$$

H_2: Starting each sub job as early as possible and choose the latest possible execution time of all unassigned jobs $I \subseteq \{1, \ldots, n\}$:

$$\mathrm{J}_k \text{ with } k = \arg\min_{i \in I} F(J_i) - d(J_i)$$

So far, a deterministic problem has been described. Machine breakdowns and the related uncertainties of occurrence of breakdowns lead to a stochastic and dynamic problem. To model the appearance of machine failures a stochastic model is necessary. This should take into account the operating time and the various demands of different jobs to the machine. In practice, a bathtub effect could be observed for the failure rate. Either the failure occurs immediately, due to defects, or later, due to wear. To reproduce this effect the Weibull distribution is used to calculate the probability that a machine breakdown occurs during the execution of a sub job. The Weibull distribution is a continuous probability distribution over the positive real numbers. It is frequently used for the study of lifetimes and failure rates. Two parameters are necessary. Scaling parameters $\lambda > 0$ is chosen according to the characteristic lifetime of a machine. The shape parameters $k > 0$ generates falling failure rates for $0 < k < 1$ and rising failure rates for $k > 1$. In order to model the bathtub effect two Weibull distributions are combined. First with shape parameter $k_1 = 0.5$ to generate falling failure rates, second with $k_2 = 5$ to generate rising failure rates. Figure 4 illustrates the failure rate generated in this way and Figure 5 the resulting cumulative distribution function of the combined Weibull distribution.

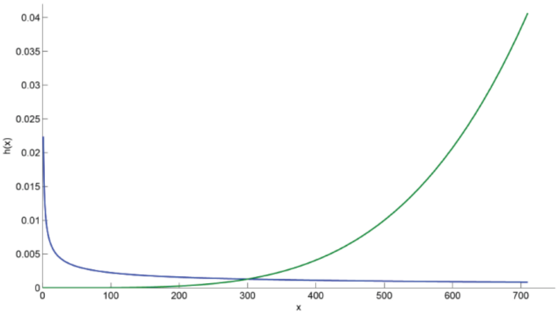

Fig. 4. Failure rates of distributions with $k_1 = 0.5$ and $k_2 = 5$ and same $\lambda = 500$.

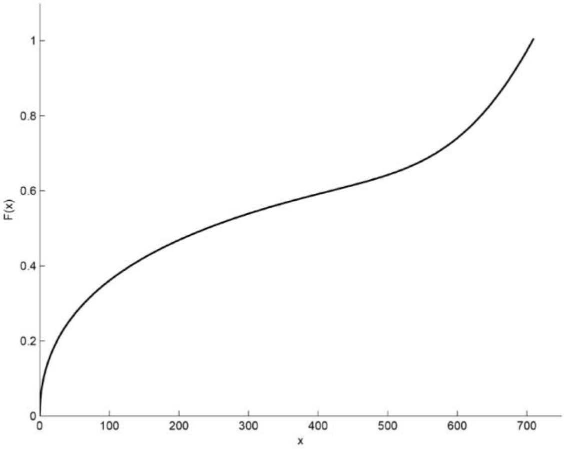

Fig. 5. Cumulative distribution function of the combined Weibull distribution.

Machine breakdowns cause interruptions in the execution of sub jobs for the duration of repairing d_r. An affected sub job can not be terminated on a different machine. Reschedule is necessary to repair not feasible plans. A repair heuristic will serve as base heuristic for the Monte Carlo Rollout algorithm. Sub jobs which have not yet started can be moved between machines to ensure all capacity constraints. Furthermore, the objective function is to be minimized.

Considering the problem as a game leads to a two persons game - the repair heuristic as first player and a random disturbance as second player. Player two starts. Thus, the stochastic dynamic $(m \times l)$ Job Shop Problem has a sequential character. This enables the use of the MCR method to determine an expected evaluation of a plan in terms of an objective function.

Starting from a plan π_0 successive, random disturbances are generated and after each disturbance a reaction using the repair heuristic follows. In detail, in each step of the random player the next machine breakdown is fixed with help of the stochastic model. The next decision step is based on the repair heuristic, which has a finite number of alternatives. The number of alternatives arises from the possible exchange options. For each alternative MCR simulations (playoffs) with search depth d are performed. So, at least after d steps, or if the end of the planning interval is reached, the playoff terminates and the objective function value is stored. For each alternative, b playoffs are played and the mean of all corresponding objective function values yield the MCR rating of this alternative. In contrast to the evaluation of the initial plan with the objective function, possible machine break downs are considered here, too. The decider chooses the alternative with the best MCR rating.

IV RESULTS AND DISCUSSION

It was considered a (2×2) Job Shop Problem with due date. Both types of machines are available twice. Each job J_i consists of two sub jobs J_{i1} and J_{i2} with $m(J_{i1}) \neq m(J_{i2})$ for all jobs $i = 1, \dots, n$. All generated instances cover n jobs and the conditions/operating hours of all machines. To build feasible plans the two heuristics H_1 and H_2 are used as well as the heuristics using the expected values $\mathbb{E}H_1$ and $\mathbb{E}H_2$. The plan with the best objective function value is used as reference.

Instance parameters have a large impact. More precisely, impacts on the success of the MCR method have: repairing time, maximal operating time of machines, initial conditions of the machines and the due date of the jobs. The first three parameters correlate with the due date. The closer the due dates are to the planned execution end times, the less room for schedule changes. At such time-critical plans almost any disturbance means a delay. The smaller the repair time, the better it is absorbed by the buffer between execution end time and due date. The greater the repair time, the greater is the resulting delay. Also the shape parameter of the Weibull distribution in the underlying stochastic model has a significant impact. As larger the shape parameters as fewer distributions occur. Also, the larger the operating hours of a machine, the more likely are the occurrence of failures.

The results demonstrate the potential of the MCR method. Significant improvements are possible, but related to search depth d and search width b. Figure 6 represent exemplary the mean distance to the best plan (created by H_1, H_2, $\mathbb{E}H_1$ or $\mathbb{E}H_2$) averaged over 1000 instances for search depths $d \leq 3000$ and different search widths, from $b = 1$ to $b = 50$. The dashed line is the mean value of the objective function by using only the heuristic.

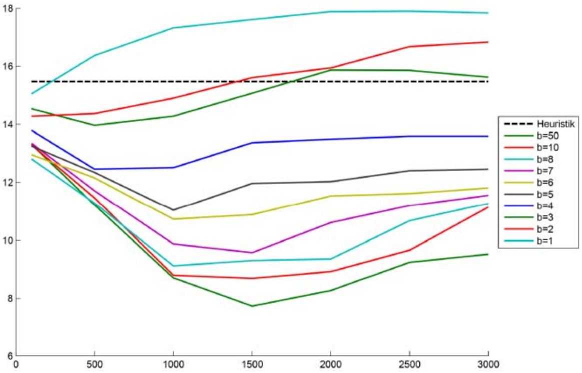

Fig. 6. Mean distance to the best plan averaged over 1000 instances with different d and b ($\lambda = 3000, d_r = 30$).

As expected, an increase of search width b results in better results. The results demonstrate that widths $b < 4$ are not practical. Worse results are obtained with increasing depth and some of them are even poorer than the heuristic results. The impact of search width is not as large as expected. So, an increase of the width is useful, but no more above a certain value b^*. In the selected model, $b^* = 20$ is independent from shape parameter λ, being seen in Figure 7. With increasing width more possible realizations of each plan are considered. Are the initial operating hours of a machine large, it is very likely that the machine failures. Due to the time-sensitive plans, the time of the failure is insignificantly, since any disruption leads to delays. Therefore a few simulations per plan are sufficed.

The search depth d is specified in time units. Only increasing the depth, without adapting the width, has a negative influence on the results, see each of the first lines in Figure 7. The greater the depth, the more gain in information about future events. But at the same time there are a much larger number of possible scenarios. Therefore, the approximation becomes worse with constant width.

The Lookahead Pathology occurs for many different parameter combinations. Here, starting with a search depth $d = 1$ an increasing depth leads first to improved results and later to poorer results, see Fig. 6. This effect is strongly dependent on the shape parameter λ of the Weibull distribution. As larger λ is as fewer disturbances occur. In this model, with a larger λ one disturbance per machine in average occurs. The search depth should not exceed the period of expected first failures. Whether there is a second failure of a machine is random according to the stochastic model. The influence of a breakdown immediately after the repairing is very large. Misinterpretations of the MCR method are likely.

Therefore, the optimal depths are $d^* = 1500$ for $\lambda = 3000$ and $d^* = 2000$ for $\lambda = 4000$. In contrast, with $\lambda = 2000$ several breakdowns per machine on average occur. In this case, a potential second breakdown is detected with increasing depth d. Here, the Lookahead Pathology does not occur.

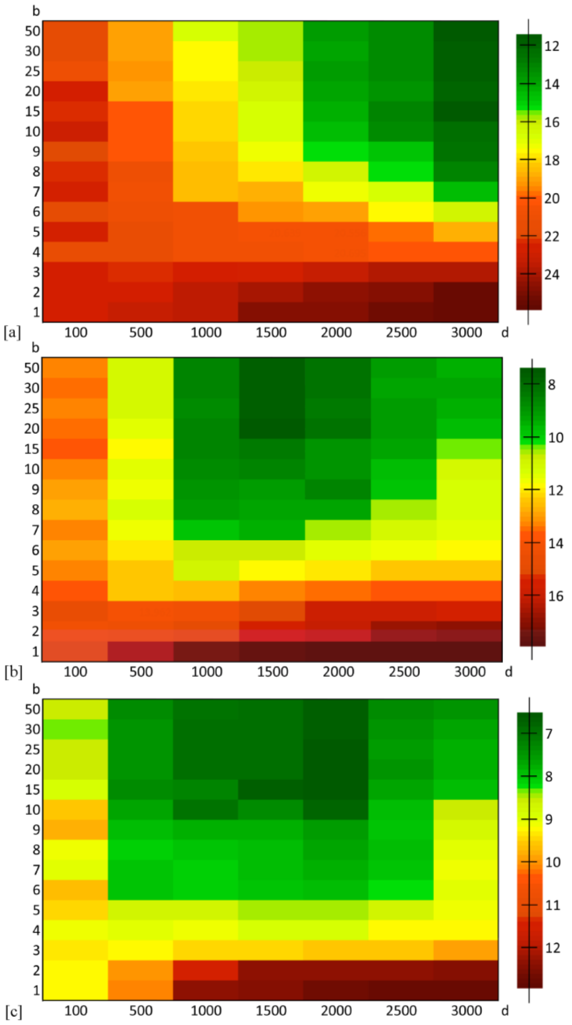

Fig. 7 Mean distance to the best plan averaged over 1000 instances with different d and b ($d_r = 30$). Shape parameter of the Weibull disturbance [a] $\lambda = 2000$, [b] $\lambda = 3000$, [c] $\lambda = 4000$.

V CONCLUSION

We have shown the potential for improvement by use of the Monte Carlo Rollout method by means of a Job Shop Problem. The results of a heuristic based on current information could be improved significantly. By using the MCR method as meta-heuristic the heuristic is combined with a stochastic model for simulating future developments. Better and more robust solutions with respect to future developments can be achieved. It has been shown, that an appropriate choice of search depth and search width is needed. The investigation has shown that increasing the width has a positive influence on the result. However, it may exist a width beyond which, the results does not improve anymore. With an increasing search depth, more information on future developments is included. But to interpret these correctly, the width must be increased, too. The Lookahead Pathology could be detected and should be considered, too.

VI ACKNOWLEDGMENTS

We acknowledge the financial support from the German Federal Ministry of Education and Research (BMBF) under the project PLUSS.

VII REFERENCES

[1] M. Rosner: Untersuchung des Monte-Carlo-Rollout-Verfahrens fuer Optimierungsprobleme unter Ungewissheit, Diplomarbeit TU Dresden, 2013.

[2] D.P. Bertsekas, J.N. Tsitsiklis, C. Wui: Rollout Algorithms for Combinatorial Optimization, Journal of Heuristics, 3, 245–262, 1997.

[3] D.P. Bertsekas, D. A. Castanon: Rollout Algorithms for Stochastic Scheduling Problems, Journal of Heuristics, 5, 89-108, 1999.

[4] G. Chaslot, S. Bakkes, I. Szita, P. Spronck: Monte-Carlo Tree Search: A New Framework for Game AI, Proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference, 5, 216-217., 2008.

[5] L. Kocsis, C. Szepesvari: Bandit based Monte-Carlo Planning, ECML-06. Number 4212 in LNCS, Springer, 5, 282-293, 2006.

[6] B. Brügman: Monte Carlo Go,1993

[7] C. Browne, E.J. Powley, D. Whitehouse, S.M. Lucas, P.I. Cowling, P.Rohlfshagen, S. Tavener, D. Perez, S.Samothrakis, S. Colton: A survey of Monte Carlo tree search methods, IEEE Trans. Comput. Intell. AI Games, 4, 1, 2012.