Optimal robot scheduling of an AGV in the RCLL and introduction of team Leuphana

Thomas Voß ✉, Jens Heger, Nicolas Meier and Anthimos Georgiadis

Leuphana University of Lüneburg,
Institute of Product and Process Innovation (PPI),
Lüneburg, Germany
{Thomas.Voss,Jens.Heger,Nicolas.Meier
Anthimos.Georgiadis}@leuphana.de,

http://www.leuphana.de/ppi

Abstract. The aim of this paper is to describe the approach on calculating the optimal schedule for an autonomous guided vehicle using a MILP (mixed integer linear program) with a solver. Considering state of the art literature a mathematical model is chosen and adopted to the given problem. Advantages and weaknesses of the method and the model are discussed. First results on the implementation are presented. Additionally a description of team Leuphana is presented.

Keywords: MILP, flexible flow shop, blocking, scheduling, AGV

1 Introduction

This paper is part of the qualification process to attend the RoboCup 2016 in Leipzig, Germany. It is organized as follows: Section 2 provides the description of the Team Leuphana and the scheduling problem in the RoboCup Logistics League (RCLL). The section describes the production scheduling problem given in the RCLL. In Section 3 a set of chosen approaches for solving the production scheduling problem with a Robotino is generally presented. Section 4 presents the first trial of a mathematical model for solving the RCLL scheduling problem. The Section 5 covers the results of the model and their discussion. In the last Section 6 the implementation of the approach is presented and evaluated. The paper closes with a future work description.

2 Team Leuphana and the research topic

Team Leuphana is a workgroup at the Institute of Product and Process Innovation (PPI) of the Leuphana University Lüneburg. The PPI is an engineering institute, which has among other fields research areas focused on intelligent systems and modeling and simulation of production processes. The transfer of research results to industry applications is strived for. The aim to participate in the RoboCup Logistics League is to develop a robot that can be integrated into a production shop floor of a company. Therefore, experts from different fields are needed, who develop a platform, which includes wiring up sensors, actors and writing the software to connect all components to a running system. Additionally strategies how to solve the production scheduling problem in the best way are needed. One of the topics Team Leuphana is working on is the optimization of the production strategy.

The RoboCup Logistics League (RCLL) is a flexible flow shop environment with an individual one piece flow. The given products ordered at dynamic release times during the game have different complexity, the simplest one called C0. All operations of the assembly have to be executed in correct order to deliver the needed product during the given time window. In the case of a C0-product the Base Station (BS) is visited once, the Cap Station (CS) is used twice, first to get the cap on a socket and second to assemble the cap to the provided base. At the last step the Delivery Station (DS) is used to deliver the product. For further details reference figure 1 and section 4. Products C1 to C3, which need intermediate products such as different colored rings, need more operations to be produced and have complex precedence constrains, regarding the mounting of rings and its cap. Due to the different routes and reentrant production processes blocking at the stations occurs. Considering the notation Graham et al. introduced [5] the RCLL can be classified as FF6,R1 |prec,r_j, d_j,$t_{jk} = t_k$,$t`_{kl}$,blocking |$\sum T_j + E_j$. The calculation of schedules has high impact on the achievement of good results, i.e. goal criteria like tardiness or overall completion time correlating to points in the RCLL.

3 Review on solving flexible shop floor scheduling problems

There are different algorithms to find a feasible schedule for the machines and the robot in the RCLL. On the one hand there are algorithms solving the problem in an optimal way and on other hand there are approximating heuristics [7]. Since the size of the RCLL scheduling problem is relatively small an optimal approach for the scheduling of the robot is investigated, also analyzing its limitations. This method is contrary to the rule based heuristic approach the world champion Carologistics uses [12]. On a general base it is known that a job shop is np-complete, but environments with 10 jobs on 10 machines are usually solvable in reasonable time. Considering the RCLL with 6 machines and 8 orders, solving the scheduling problem might be possible for this case.

The scheduling problem for job and flow shops has been realized with MILP (mixed integer linear programming) by many researchers during the last years. An overview on different approaches solving a flexible job shop scheduling problem (FJSSP) is given by Chaudhry in 2016. Recognizable is also the fact, that the most commen criteria for performance measure are make span of an order or workload of machines [2]. Brucker and Knust have shown that problems in the category $F2\ |t_j\ |\sum C_j$ are solvable in polynomial time under certain circumstances. It is also shown that $F2\ |t_j{,}r_j\ |\sum C_j$ is NP-hard and no longer solvable in polynomial time [1]. Still a general solution for flexible job shop (FJSSP) environments is given by Özgüven et al. [13].

Blocking in a JSSP describes the absence of a storage capacity between machines. Mascis and Pacciarelli developed a model to simulate the blocking constrain in 2001 [10]. Ronconi developed models considering the minimization of earliness and tardiness of an order under the restriction of blocking in 2012 . This optimization goal is related to JIT-production, where a certain time window has to be met [15] and can be used to get a optimal result for the RCLL.

The problem of supplying material to machines by AGVs is called part-feeding. Although part feeding is well known by car manufacturers with high product flexibility [9] and part-feeding-problems in assembly-lines in the electronic industry, scheduling AGVs that can drive individually is not applied often. Different approaches consider tow-trains with given routes (on rails or on the ground) with central storage units (supermarket) [4], individual transport systems with central supply units and given routes [16] as well as individual robots with a central storage unit (bartender) and no route defined [3, 11]. The last one given being equivalent to the RCLL.

4 Approach

Considering that the robots do not differentiate from another, every robot can do every task, makes them identical parallel. To simplify the modeling of the shop floor, the robot is categorized as a regular machine and the transportation time is considered as processing time. This leads to the fact that the problem can be considered FF7 |prec, r_j, d_j, blocking $|\sum T_j + E_j$ and can be solved like a blocking flexible flow shop problem with reentrant processes. The Robotino can be seen and used as a central hub machine. The existence of a hub is given if a job being processed on other machines between any two consecutive entries into the same machine (the hub). [8].

The model of a job shop [14] has to be adjusted according to the requirements of the application, in our case the RCLL. Given a set of JOBS, MACHINES and OPerations: i being Element in JOBS, j describes the specific OPeration on MACHINE k. The parameter r[i,j,k] defines if process j of order i is processed on machine k as binary. Z[i,j,k] describes if the Operation j of order i is ahead of operation jj of order i and ii as binary. Parameter p[i,j,k] describes the process time of operation j of order i on machine k. H is a big positive number. Variable s[i,j] describes the start time of operation j of order i.

Equation 1 describes the starting time of an operation being bigger than the starting time of the precedence operation plus its process time. The equations 2 and 3 represent the relationship between the operations of the available orders and restrict one machine to processing a single operation at a time. Equation 4 (also called R5) minimizes the completion time for all orders.

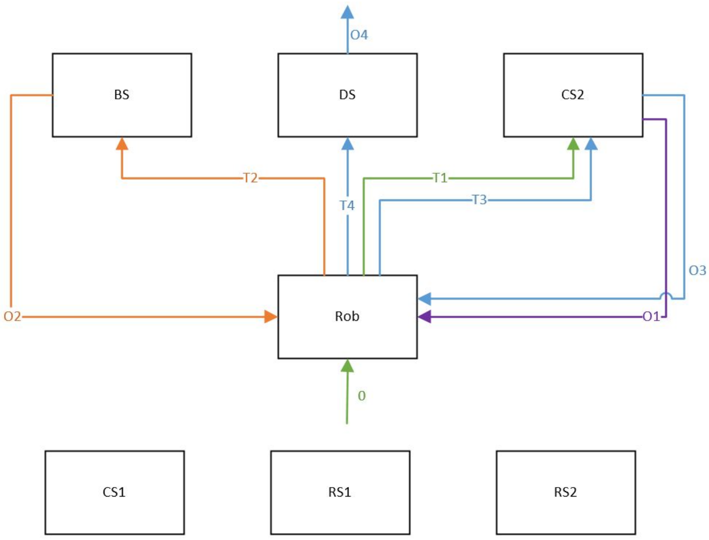

Fig. 1. The Order C0 as a flexible flow shopu with reentrant operations

$$R2: \sum_{k=1}^{m} r_{i,j,k} * (s_{i,j} + p_{i,j,k}) \leq \sum_{k=1}^{m} r_{i,j+1,k} * s_{i,j+1} \qquad (1)$$
$$i = 1, .., n; j = 1, ..N_i - 1;$$

$$R3: H(2 - r_{i,j,k} - r_{ii,jj,k}) + H(1 - Z_{i,ii,j,jj}) + (s_{ii,jj} - s_{i,j}) \geq p_{i,j,k} \qquad (2)$$
$$1 \leq i \leq ii \leq n; j = 1, ..N_i, jj = 1, ..N_{ii}; k = 1, .., m;$$

$$R4: H(2 - r_{i,j,k} - r_{ii,jj,k}) + H * Z_{i,ii,j,jj}) + (s_{i,j} - s_{ii,jj}) \geq p_{ii,jj,k} \qquad (3)$$
$$1 \leq i \leq ii \leq n; j = 1, ..N_i, jj = 1, ..N_{ii}; k = 1, .., m;$$

$$R5: \sum_{k=1}^{m} r_{i,N_i,k} * (s_{i,N_i} + p_{i,N_i,k}) \leq Cmax \qquad (4)$$
$$i = 1, .., n;$$

5 Results and Discussion

The model was solved with Gurobi 6.5.1. on a Intel®Core i7-4710HQ CPU with 2.5 GHz and 8 GB RAM. This being only slightly faster than the Robotino which contains an Intel®Core i5 with 2.4 GHz and 8 GB RAM. First tests have shown that solving the scheduling of four C0 Orders takes less than 1 second. Tests have also shown, that doubling the amount of C0-orders from two to four leads to a rise of restriction rows Gurobi had to solve by factor five.

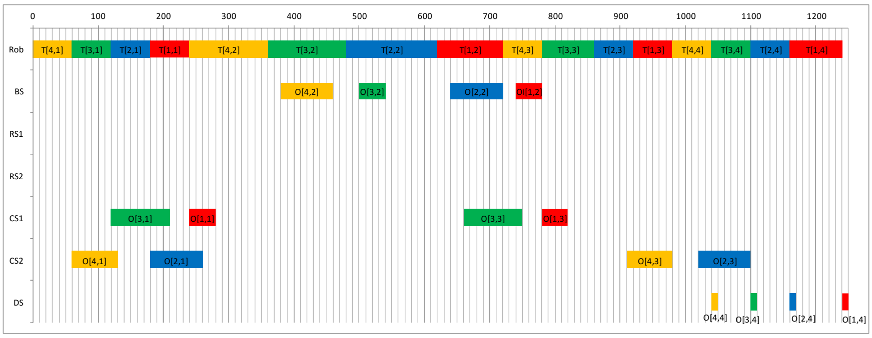

Fig. 2. A feasible solution for scheduling 4 C0 Orders

The solution to the given restrictions provided by the solver contains a feasible schedule for all machines in the model. In Figure 2 a general solution to the job shop scheduling problem is given. [1]

In the figure 2 it can be seen that Operation O[4,2] does not start at the earliest possible time, right after Operation T[4,2] finished. The reason is the optimization criteria defined for the solver. Regardless of the starting time of the machine operation, the robot is used to 100 % and there is no possible solution to finish the task of the robot any earlier. So there is no reason the change the starting time of O[4,2], if all restrictions are met. This could be eliminated by adding an additional term to the goal function in the MILP. Due to the complexity of the products the aspects of blocking, rescheduling, parallel processing and assembly have to be added to the model to suit the RCLL.

[1] All data and model files as well as the general MPS file will be available at Link https://bitbucket.org/robotinoentwicklung/robotino3/wiki/Home

6 Conclusion

However, it has to be evaluated if the scheduling of more complex orders is still feasible in a short time. Preliminary result show, that the integration of the mathematical model is possible in the context of the RCLL. Different optimization criteria and goal functions have to be tested for an optimal solution. The results generated by reduction of workload on the robot and the impact of adding more robots have to be assessed and documented for further research.

If the model turns out to be to complex or scheduling while operating the robot is not possible due to a lack of CPU and RAM, an abort-criteria has to be defined. Since, this approach is not real-time compliant, additional fall-back procedures need to be implement. Furthermore, dynamic changes and unforeseen occurring events need to be handled by a rescheduling method, e.g., priority rules. The MILP can also be used to evaluate the performance of other heuristics, e.g. priority rules, or help to develop dynamic adaption approach [6].

The implementation on the robot is going to be realized in Matlab Mathworks and the ROS interface provided by Mathworks. A state-machine will get the needed input to handle the tasks in the given order.

References

[1] Brucker, P., Knust, S., Cheng, T., Shakhlevich, N.: Complexity results for flow-shop and open-shop scheduling problems with transportation delays. Annals of Operations Research 129(1-4), 81–106 (2004)

[2] Chaudhry, I.A., Khan, A.A.: A research survey: Review of flexible job shop scheduling techniques. International Transactions in Operational Research 23(3), 551–591 (2016)

[3] Dang, Q.V., Nielsen, I., Steger-Jensen, K., Madsen, O.: Scheduling a single mobile robot for part-feeding tasks of production lines. Journal of Intelligent Manufacturing 25(6), 1271–1287 (2014)

[4] Fathi, M., Alvarez, M.J., Hassani Mehraban, F., Rodríguez, V.: A multiobjective optimization algorithm to solve the part feeding problem in mixed-model assembly lines. Mathematical Problems in Engineering 2014(1), 1–12 (2014)

[5] Graham, R.L., Lawler, E.L., Lenstra, J.K., Kan, A.: Optimization and approximation in deterministic sequencing and scheduling: a survey. In: North-Holland Publishing Company (ed.) Annals of Discrete Mathematics, vol. 5, pp. 287–326 (1979)

[6] Heger, J., Branke, J., Hildebrandt, T., Scholz-Reiter, B.: Dynamic adjustment of dispatching rule parameters in flow shops with sequence-dependent set-up times. International Journal of Production Research pp. 1–13 (2016)

[7] Jain, A., Meeran, S.: A state-of-the-art review of job-shop scheduling techniques. European Journal of Operations Research (113), 390–434 (1999)

[8] Kubiak, W., Sheldon X. C. Lou, Wang, Y.: Mean flow time minimization in reentrant job shops with a hub. Operations Research 44(5), 764–776 (1996)

[9] Lee, K.M.: Flexible part-feeding system for machine loading and assembly. part i. a state-of-the-art survey. International Journal of Production Economics 25(1–3), 141–153 (1991)

[10] Mascis, A., Pacciarelli, D.: Job-shop scheduling with blocking and no-wait constraints. European Journal of Operational Research 143(3), 498–517 (2002)

[11] Nielsen, I., Dang, Q.V., Bocewicz, G., Banaszak, Z.: A methodology for implementation of mobile robot in adaptive manufacturing environments (2015)

[12] Niemueller, T., Reuter, S., Ewert, D., Ferrein, A., Jeschke, S., Lakemeyer, G.: The carologistics approach to cope with the increased complexity and new challenges of the robocup logistics league 2015. In: RoboCup 2015: Robot World Cup XIX, pp. 47–59. Springer (2015)

[13] Özgüven, C., Özbakır, L., Yavuz, Y.: Mathematical models for job-shop scheduling problems with routing and process plan flexibility. Applied Mathematical Modelling 34(6), 1539–1548 (2010)

[14] Pan, J.C.H., Chen, J.S.: Mixed binary integer programming formulations for the reentrant job shop scheduling problem. Computers & Operations Research 32(5), 1197–1212 (2005)

[15] Ronconi, D.P., Birgin, E.G.: Mixed-integer programming models for flowshop scheduling problems minimizing the total earliness and tardiness. In: Just-in-Time systems, pp. 91–105. Springer (2012)

[16] Singh, N., Sarngadharan, P.V., Pal, P.K.: Agv scheduling for automated material distribution: a case study. Journal of Intelligent Manufacturing 22(2), 219–228 (2009)