Real-World Shop Floor Scheduling by Ant Colony Optimization

Andre Vogel, Marco Fischer, Hendrik Jaehn, and Tobias Teich

Chemnitz University of Technology, Dept. of Economics, Chair BWL VII*
D-09107 Chemnitz, Germany
{andre.vogel,marco.fischer,hendrik.jaehn,t.teich}
@wirtschaft.tu-chemnitz.de

Abstract. Manufacturing Control Problems are still often solved by manual scheduling, that means only out of the workers experience. Modern algorithms, such as Ant Colony Optimization, have proved their capacity to solve this kind of problems. Nevertheless, they are only used exceptionally in real world. There are two main reasons for that. Firstly, an ant-based scheduling tool has to fit into the organizational structures of today's companies, i.e. it has to be coupled with the Enterprise Resource Planning-system (ERP-system) used in the company, in order to ensure that the capacity of the colonies search is used as efficiently as possible. The second reason is the size of the real world shop floor scheduling problems. In order to be able to deal with that problem, the authors propose a continuously operating Ant Algorithm, which can easily adapt to sudden changes in the production system.

1 Introduction to the Problem

Shop floor scheduling solves the machine allocation problem. It is a much examined, $\mathcal{NP}$-hard problem.

In Operations Research, models for these kind of problems and heuristics to solve them had been developed. But it was still not possible to handle shop floor scheduling as an integral whole. The situation changed firstly due to the introduction of evolutionary methods and other meta-heuristics like Simulated Annealing or Treshold Accepting. A number of attempts have been made to solve hard combinatorial optimization problems in the field of production control using Genetic Algorithms [2], [13].

The objective of this work was to compare the developed Ant Algorithm - which will be described in the third section - to the results achieved by a Genetic Algorithm tested on the same real-world data.

2 The Ant Algorithm

The used algorithm, which is based on several works on this topic ([1], [3], [4], [6] and [11]), has already been tested successfully e.g. in order to deal with the widely

* phone: ++49-371-531 42 52

M. Dorigo et al. (Eds.): ANTS 2002, LNCS 2463, pp. 268–273, 2002.

known Job Shop Scheduling Benchmark problems [9]. Job Shop Scheduling is closely related to the problem considered here.

In order to use the algorithm also for Real-World Shop Floor Scheduling problems, some adaptions and changes within the algorithm are necessary.

To store the pheromone, a Position-Operation-Pheromone-matrix (P-O-P-matrix) is employed (see figure 1). Within that, the operations are stored by an accessary operation number (onr). Pheromone values ($\tau_{pos,onr}$) are allocated to the fields of the matrix during the solution process of the algorithm. The pheromone values of an operation depend on the position within the solution sequence (permutation). The pheromone values are changed by the pheromone update within each iteration according to the new solutions.

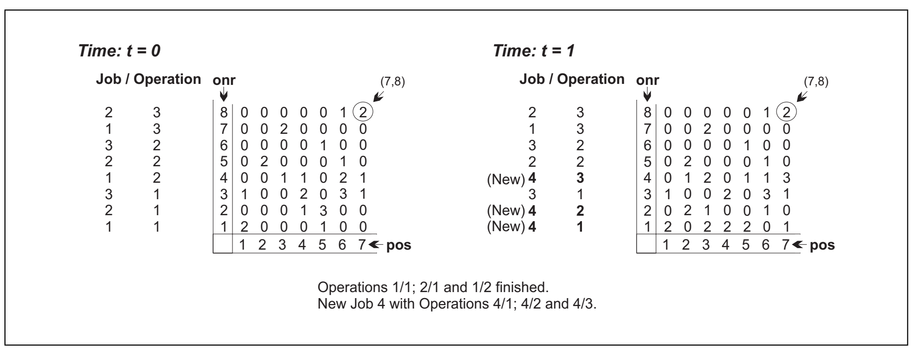

Fig. 1. Position-operation-pheromone-matrix

If a costumer inquiry is considered, a copy of the problem is used to generate an answer. A new job is set up after the costumer had accepted an offer. The new data is handed over to the Ant Colonies (by using an XML-File[12]) , which are updating the P-O-P-matrix and start the search again.

Due to the filling up of the gaps, which are result of the finished jobs, and the dynamic structure of the P-O-P-matrix connected with that, an allocation table is required (see figure 2), which establishes the relationship between the operations within a job and the operation number (onr).

It is necessary to reset the pheromone values due to the continuously incoming jobs/operations that have to be scheduled.

In case, a best solution is found in the end, all the operations of that solution receive a certain additional amount of pheromone, appropriate to their position in the solution. Because of that, during the next search, it is a lot more attractive to schedule that operation again in the position because that position of the operation already proved success. Thereby, the amount of the pheromone that has to be delivered again, depends on two factors. On the one hand, it depends on the time that remains between the current time (tc) and the attainment of the delivery date (td_j). That means, operations within a job having an earlier

Allocation Table (t=1)

Job (J_n)	Operation (O_j)	Operation-number (onr)
1	3	7
2	2	5
	3	8
3	1	3
	2	6
4	1	1
	2	2
	3	4

Fig. 2. Allocation table (time: t=1)

delivery date are considered more important in scheduling than those operations of jobs having a relatively greater temporal buffer (tf_j).

$$tf_j = te_j - tc \tag{1}$$

The end of the job time window (te_j) results out of the difference between the delivery date (td_j) and the sum of the processing durations of the remaining operations of a job (see equation 2).

$$te_j = td_j - \sum_{k \in O_{jk}} p_{jk} \tag{2}$$

$$O_{jk} \in \mathcal{PO}$$

A second factor is the necessity of the increase of jobs with a higher priority, and wit that of the operations belonging to these jobs, in comparison to other jobs: This is considered within the initial pheromone values.

In case, the P-O-P-matrix has been developed and the initial values of the pheromone are initialized, the algorithm starts. Each ant chooses the next operation (onr), according to the set of feasible operations ($\mathcal{PO}$), which can be processed in the next step, as follows:

$$p_{pos,onr} = \frac{[\tau_{pos,onr}]^{\alpha} \cdot [\eta_{pos,onr}]^{\beta}}{\sum_{h \in \mathcal{PO}} [\tau_{pos,h}]^{\alpha} \cdot [\eta_{pos,h}]^{\beta}} \tag{3}$$

$$p_{pos,onr} = \frac{(\sum_{k=1}^{pos} [\gamma^{pos-k} \tau_{k,onr}])^{\alpha} \cdot [\eta_{pos,onr}]^{\beta}}{\sum_{h \in \mathcal{PO}} (\sum_{k=1}^{pos} [\gamma^{pos-k} \cdot \tau_{k,h}])^{\alpha} \cdot [\eta_{pos,h}]^{\beta}} \tag{4}$$

$$\eta_{pos,onr} = \frac{1}{tf_j^{\frac{\sum_j tf_j}{(tf_j)^q}}} \tag{5}$$

$$tf_j = te_j - tb_j \tag{6}$$

In order to be offered a detailled description of the used equations (3) and (4), consider [8] und [5].

Within the solution search, the heuristics value ($\eta_{pos,onr}$) is calculated with regard to equation (5) in the following. The effect of the urgency on the selection can be influenced by the parameter q. The greater q the higher is the probabilitiy of the selection of an operation disposing of a quite small temporal buffer.

The next technologically possible operation of a job is part of $\mathcal{PO}$, that means, part of the quantity of the operations that can be scheduled. The start of the time window of the following operations (tb_j) of a job results from the present time (tc), because the previous operations have already been completed and the next possible operations ($onr \in \mathcal{PO}$) of a job could theoretically be scheduled immediately:

$$tb_j = tc$$

The amount of the pheromone of an operation at a position is restricted by certain upper- (τ_{max}) and lower limits (τ_{min}).

$$\tau_{min} < \tau_{pos,onr} < \tau_{max}$$

This idea is based on the works of Stuetzle [10] within which a higher solution quality for known benchmark-problems has been achieved.

After all ants of all colonies generated a solution in this way, the global pheromone update is carried out. Therefore, an elitist-strategy is applied. This means that the ant having generated the best solution (mean lateness L) is allowed to add pheromone according to:

$$\tau_{pos,onr} = \tau_{pos,onr} + \rho * \frac{1}{L}$$

In this case, pheromone is added to the positions of an operation in the current best solution within the P-O-P-matrix.

After every generation, the evaporation is carried out in every cell of the matrix, as follows:

$$\tau_{pos,onr} = (1 - \rho) \cdot \tau_{pos,onr} \tag{7}$$

Parameter ρ regulates the evaporation rate. The smaller ρ is, the longer the pheromone informations are passed on.

3 Computational Results

The shop floor scheduling problem was tested on real world data provided by a German engineering company. The plant is organized according to the workshop principle, which means that machines are grouped according to the operations that have to be carried out using this ressource rather than according to the products that are manufactured using this machines.

In order to be able to dispose of real world problem sizes and data, the following numbers were recorded within a period of two months. The release date, the finishing date and the due date of each job, as well as the date and the duration of all occured disturbances in the manufacturing process (machine breakdowns, material defects, illnesses of workers, etc.) were observed.

Because of comparing the reached solution quality, we used the mean flow time and the mean lateness of jobs. For the comparison of the results achieved by the means of the Ant Algorithm and other methods we used a priority rule scheduler (PRIORULE) and GACOPA which is a Genetic Algorithm using continuous co-evolution of the parameter settings and parallelization [7]. The priority rule scheduler was allowed to calculate a number of single, standard priority rules (e.g.: shortest processing time, earliest due dates etc.) and combinations of these rules. The best out of the achieved results was used. Manual scheduling achieved a mean lateness of 2.4 days and a mean flow time of 13.2 days. These values were set to 100% to ease the comparison.

Table 1. Relative results (manual scheduling = 100%)

scheduler	*mean lateness*	*mean flowtime*
MANUAL	100.0 %	100.0 %
PRIORULE	95.3 %	97.9 %
GACOPA	55.3 %	85.0 %
ANTS	61.8 %	87.7 %

The following parameters were employed for the tests: $\alpha = 1; \beta = 1; \rho = 0,01; c = 0,8; \gamma = 0,8; q = 2$. The results presented in table 1 show that ants perform very well, but not as good as the GACOPA, the results are compared with.

4 Conclusion

As a result it can be stated that modern scheduling tools for real world shop floor scheduling allow companies to mobilize a lot of reserves in the production system at comparably low cost. The quality of the GA solution wasn't achieved, but it was the first work completed in this field with ACO. The authors further research will be directed towards improving the presented approach by means of solution quality, reducing of computation time and improving the parameter settings.

Acknowledgements. This work has been supported by the Collaborative Research Center 457 and the CBS GmbH.

References

1. Bonabeau, E. Dorigo, M. Theraulaz, G.: Swarm Intelligence - From Natural to Artificial Systems, pp. 69-71, Oxford University Press, New York, NJ, 1999.
2. Davis, L.: Job Shop Scheduling with Genetic Algorithms. in: Grefenstette, J., Editor, Proceedings of an International Conference on Genetic Algorithms and their Applications, pp. 136-140, Hillsdale, Lawrence Erlbaum Associates, 1985.
3. Deneubourg, J.-L., et. al: Self-Organization Mechanisms in Ant Societies (II): Learning in Foraging and Division of Labour. Experientis Suppl. 54, pp. 177-196, 1987.
4. Dorigo, M.: The Ant System: Optimization by a colony of cooperating agents, in: IEEE Transactions on Systems, Man, and Cybernetics 26, (1), pp.29-41, 1996.
5. Fischer, M., Vogel, A., Teich, T., Fischer, J.: A new Ant Colony Algorithm for the Job Shop Scheduling Problem. in: Proceedings of the Genetic and Evolutionary Computation Conference, San Francisco, 2001.
6. Gambardella, L.M., Agazzi, G.: MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows, in: Corne, D., Dorigo, M., Glover, F.(Eds.) - New Ideas in Optimization, pp.63-76, 1999.
7. Kaeschel, J., Meier, B., Fischer, M., Teich, T.: Real-World Applications: Evolutionary Real World Shop Floor Scheduling using Parallelization and Parameter Coevolution, in: Proceedings of the Genetic and Evolutionary Computation Conference, Las Vegas, 2000.
8. Merkle, D., Middendorf, M.: An Ant Algorithm with Global Pheromone Evaluation for Scheduling a Single Machine , in: Cagnoni, S., et al. (Eds.) - Real-World Applications of Evolutionary Computing, Proceedings of EvoWorkshops 2000, Edinburgh, LNCS 1803, pp.281-296.
9. OR-library: http://graph.ms.ic.ac.uk/info.html, 2002.
10. Stuetzle, T., Hoos, H.: MAX-MIN Ant System for the Traveling Salesman Problem, 1997.
11. van der Zwaan, S., Marques, C.: Ant colony optimization for job shop scheduling. in: Proceedings of the Third Workshop on Genetic Algorithms and Artificial Life (GAAL 99), 1999.
12. World Wide Web Consortium (Eds.): Extensible Markup Language (XML), http://www.w3.org/XML, 24.03.2001.
13. Yamada, T., Nakano, R.: A Genetic Algorithm with Multi-Step Crossover for Job Shop Scheduling Problems. in: Proceedings of an International Conference on GAs in Engineering Systems: Innovations and Applications (GALESIA 1995), pp. 146-151, 1995.