



Multi-objective Genetic Algorithm for Real-World Mobile Robot Scheduling Problem

Quang-Vinh Dang, Izabela Nielsen, Kenn Steger-Jensen

► To cite this version:

Quang-Vinh Dang, Izabela Nielsen, Kenn Steger-Jensen. Multi-objective Genetic Algorithm for Real-World Mobile Robot Scheduling Problem. 19th Advances in Production Management Systems (APMS), Sep 2012, Rhodes, Greece. pp.518-525, 10.1007/978-3-642-40352-1_65 . hal-01472286

HAL Id: hal-01472286

<https://inria.hal.science/hal-01472286>

Submitted on 20 Feb 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Multi-Objective Genetic Algorithm for Real-World Mobile Robot Scheduling Problem

Quang-Vinh Dang¹, Izabela Nielsen¹, Kenn Steger-Jensen¹

¹Department of Mechanical and Manufacturing Engineering, Aalborg University,
Fibigerstræde 16, 9220 Aalborg, Denmark
{vinhise, izabela, kenn}@m-tech.aau.dk

Abstract. This paper deals with the problem of scheduling feeding tasks of a single mobile robot which has capability of supplying parts to feeders on production lines. The performance criterion is to minimize the total traveling time of the robot and the total tardiness of the feeding tasks being scheduled, simultaneously. In operation, the feeders have to be replenished a number of times so as to maintain the manufacture of products during a planning horizon. A method based on predefined characteristics of the feeders is presented to generate dynamic time windows of the feeding tasks which are dependent on starting times of previous replenishment. A heuristic based on genetic algorithm which could be used to produce schedules in online production mode is proposed to quickly obtain efficient solutions. Several numerical examples are conducted to demonstrate results of the proposed approach.

Keywords: Multi-objective, Scheduling, Mobile Robot, Genetic Algorithm

1 Introduction

The automation technology in combination with advances in production management has dramatically changed the equipment used by manufacturing companies as well as the issues in planning and control. With these changes, highly automated and unmanned production systems have become more popular in several industrial areas, e.g., automotive, robot, and pump manufacturing [3]. An automatic production system consists of intelligent and flexible machines and mobile robots grouped into cells in such a way that entire production of each product can be performed within one of the cells. With embedded batteries and manipulation arms, mobile robots are capable of performing various tasks such as transporting and feeding materials, tending machines, pre-assembling, or inspecting quality at different workstations. They have been thus employed in not only small companies which focus on exact applications and a small range of products, but also large companies which can diversify applications in a longer term and larger range. Within the scope of this study, a given problem is particularly considered for a single mobile robot which will automate part-feeding tasks by not only transporting but also collecting containers of parts and emptying them into the feeders needed. However, to utilize mobile robots in an efficient

manner requires the ability to properly schedule feeding tasks. Hence, it is important to plan in which sequence mobile robots process feeding operations so that they could effectively work while satisfying a number of practical constraints.

The problem of scheduling part-feeding tasks of the mobile robot has been modeled in some respects comparable to the Asymmetric Traveling Salesman Problem (ATSP) which belongs to the class of NP-hard combinatorial optimization problems [7]. Among heuristic approaches, Genetic Algorithm (GA) has been widely used in the research areas of TSP, ATSP, or robot task-sequencing problems. Liu and Zheng [10], Moon et al. [12], and Snyder and Daskin [13] discussed about using GAs to solve TSP, while Choi et al. [2] and Xing et al. [14] proposed GAs to deal with ATSP. Zacharia and Aspragathos [15] introduced a method based on GA and an innovative encoding to determine the optimal sequence of manipulator's task points which is considered an extension to the TSP. Beside genetic algorithms, Bocewicz [1] presented the knowledge-based and constraint programming-driven methodology in planning and scheduling of multi-robot in a multi-product job shop taking into account imprecise activity specifications and resource sharing. Hurink and Knust [9] proposed a tabu search algorithm for scheduling a single robot in a job-shop environment considering time windows and additionally generalized precedence constraints. Maimon et al. [11] also presented a neural network approach with successful implementation for the robot task-sequencing problem.

Although there are many related research, the problem of scheduling a single mobile robot with dynamic time windows and restricted capacity where multiple routes have to be carried out has surprisingly received little attention in the literature despite its important applications in practice, e.g. part-feeding task. Such a task must be executed a number of times within time windows which are dependent on starting times of the previous executions of that task, hence, the term, dynamic time windows. The objectives of minimizing the total traveling time of the robot and the total tardiness of the tasks are taken into account to support the global objective of maximizing system throughput. The existing approaches are not well suited and cannot be directly used to solve the problem. Thus, in this paper, a heuristic based on GA, a possibly promising approach to the class of multi-objective optimization, is developed to find efficient solutions for the problem. The advantageous feature of GA is the multiple directional and global search by maintaining a population of potential solutions from generation to generation. Such population-to-population approach is useful to explore all non-dominated solutions of the problem [6].

The remainder of this paper is organized as follows: in the next section, problem statement is described while a genetic algorithm-based heuristic is presented in Section 3. Numerical examples are conducted to demonstrate results of the proposed approach in Section 4. Finally, conclusions are drawn in Section 5.

2 Problem Statement

The work is developed for a cell which produces parts or components for the pump manufacturing industry at a factory in Denmark. The essential elements considered in

the manufacturing cell consist of an autonomous mobile robot with limitation on carrying capacity, a central warehouse designed to store small load carriers (SLCs), and multiple feeders designed to automatically feed parts to machines of production lines. Besides, every feeder has three main characteristics including maximum level, minimum level, and part-feeding rate to machine. In operation, the robot will retrieve and carry one or several SLCs containing parts from the warehouse, move to feeder locations, empty all parts inside SLCs, then return to the warehouse to unload empty SLCs and load filled ones. To maintain the manufacture of a quantity of products during a given planning horizon, the feeders (tasks) have to be replenished a number of times, the robot consequently has a set of subtasks of tasks to be carried out within time windows. Such a time window of a subtask of a task could be only determined after starting time of the previous subtask of that task. Fig. 1 below shows a layout of the described manufacturing cell.

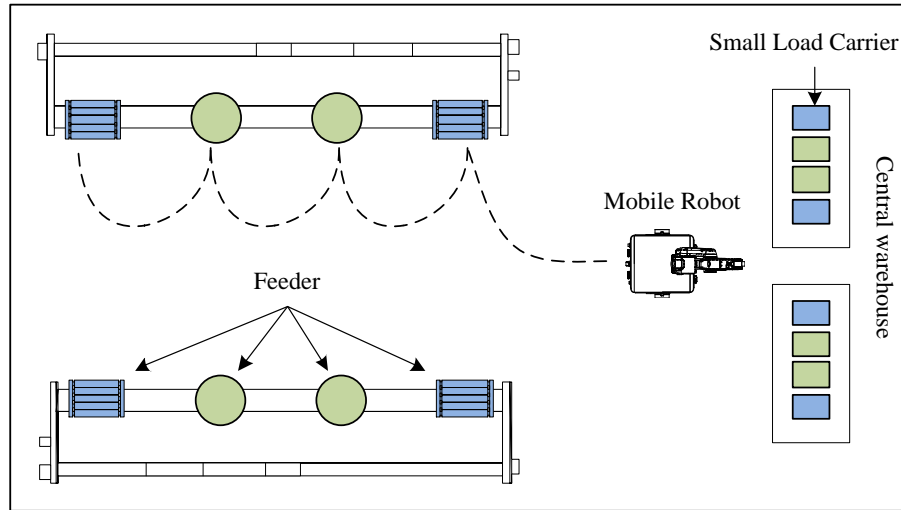


Fig. 1. Layout of the manufacturing cell

To enable the construction of a feeding schedule for the mobile robot, assumptions are considered as follows:

- The robot can carry one or several SLC(s) at a time.
- All tasks are periodic, independent, and assigned to the same robot.
- Working time, traveling time between any pairs of locations of the robot, and part-feeding rate to machine of a feeder are known.
- All feeders of machines must be fed up to maximum levels and the robot starts from the ware house at the initial stage.

In order to accomplish all the movements with a smallest consumed mount of battery energy, the total traveling time of the robot is an important objective to be considered. Apart from that, another performance measure is the amount of time a feeder has been

waiting to be replenished by the robot. Alternatively, due time of a time window of a feeding task could be considered soft constraint, i.e. schedules that do not meet this constraint are taken into account. In addition, making decisions on which way the robot should provide parts to feeders is a part of real-time operations of production planners. Moreover, concerning the problem belong to NP-hard class, computation time exponentially grows with the size of the problem (e.g. larger number of feeders). It is therefore necessary to develop a computationally effective algorithm, namely GA-based heuristic, which determines in which sequence the feeders should be supplied so as to minimize the total traveling time of the robot and the total tardiness of feeding tasks while satisfying a number of practical constraints.

3 Genetic Algorithm-based Heuristic

In this section, genetic algorithm, a random search method taking over the principle of biological evolution [8], is applied to develop a heuristic which is allowed to convert the aforementioned problem to the way that efficient solutions could be found. The GA-based heuristic shown in Fig. 2 comprises of the following components: genetic representation and initialization; constraint handling and fitness assignment; genetic operators including selection, crossover, and mutation; termination criteria.

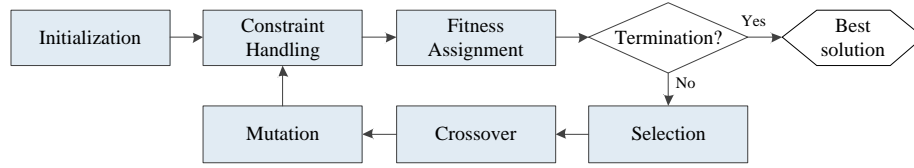


Fig. 2. Flow chart of GA-based heuristic

3.1 Genetic Representation and Initialization

For the problem under consideration, a solution can be represented by a chromosome of non-negative integers $(0, i, j, \dots, i, \dots, k)$ which is an ordering of part-feeding tasks of the robot where i, j, k : feeder index; $i, j, k = 1 \div n$; n : number of feeders. The original length of a chromosome is equal to the total number of subtasks of tasks added the first subtask of task at the central warehouse $(1 + \sum n_i; n_i$: number of subtasks of task $i)$.

For the initial generation, genes on a chromosome are randomly filled with tasks at feeders. The frequency of such a task is the number of subtasks of that task, in other words, number of times that tasks has to be executed.

3.2 Constraint Handling and Fitness Assignment

After initialization or crossover and mutation operations, chromosomes are handled to be valid and then assigned fitness values. A valid chromosome should satisfy two

constraints of limitation on carrying capacity Q_m of the robot and time windows of subtasks of part-feeding tasks. For the first type of constraints, to guarantee the robot not to serve more number of feeders than number of SLCs carried in one route, the subtasks of task at the warehouse represented by zeroes are inserted into a chromosome after every Q_m genes starting from the first gene. For instances, if the limitation on carrying capacity of the robot is two SLCs of parts, the chromosome should be restructured to be $(0, i, j, 0, \dots, 0, k, l, 0, \dots, 0)$.

The second type of constraints requires a subtask of a task to be started after release time and completed by the due time of that subtask, if possible. As mentioned, due time constraints are considered soft constraints. They thus could be modeled as an objective of the total tardiness of part-feeding tasks. The release time and due time could be determined as shown in Equation (1) and (2) below.

$$r_{ik+1} = s_{ik} + (u_i - v_i) \times c_i, i = 1 \div n, k = 1 \div n_i \quad (1)$$

$$d_{ik} = r_{ik} + (v_i \times c_i), i = 1 \div n, k = 1 \div n_i \quad (2)$$

where r_{ik}, d_{ik}, s_{ik} : release time, due time, and starting time of subtask k of task i
 u_i, v_i, c_i : maximum level, minimum level of parts of feeder i , and part-feeding rate to machine of feeder i

After constraint handling procedure, the objectives of the total traveling time of the robot and the total tardiness of part-feeding tasks are calculated one after another for every chromosome in the population. A weighted-sum fitness function F is then used to assign a fitness value to each chromosome as shown in Equation (3) where t_{ij} is traveling time of the robot from one location to another, w_i is working time of the robot per SLC at feeder i , and α is the weighted coefficient.

$$F = \alpha \times (\sum_{i,j} t_{ij}) + (1 - \alpha) \times (\sum_{i,k} (\max\{0, (s_{ik} + w_i/2) - d_{ik}\})) \quad (3)$$

3.3 Genetic operators

Selection, crossover, and mutation are three main genetic operators. For selection, various evolutionary methods could be applied in this problem. $(\mu + \lambda)$ selection is used to choose chromosomes for reproduction. Such selection mechanism guarantees that the best solutions up to now are always in the parent generation [4-5].

Crossover operator generates offspring by combining the information contained the parent chromosomes so that the offspring inherits good features from their parents. The Roulette-wheel selection is used to select the parent chromosomes based on their weighted-sum fitness values. Order crossover (OX) [6] operated with probability P_c will be employed to generate an offspring as follows. Genes having zero values are removed before two cut points are randomly chosen on the parent chromosomes. A string between these cut points in one of the parent chromosomes is first copied to the offspring, the remaining positions are then filled according to the sequence of genes in the other parent starting after the second cut point. When an offspring is produced, it undergoes insertion mutation [6] with probability P_m which selects a gene at random and inserts it in a random position.

3.4 Termination Criteria

Termination criteria are employed to determine when the GA-based heuristic should be stopped. Note that making decisions on which sequences the robot should serve feeders is a part of real-time operations of production planners. Therefore, on the one hand if the best solutions over generations do not converge to a value, the maximum generation G_m would be used to stop the run. On the other hand, if the best solution does not improve over G_c consecutive generations, it would not be valuable to continue searching.

4 Numerical Examples

The performance of the GA-based heuristic will be tested on several problem instances in this section. Three problems, which are as similar to the real-world case as they can be, are generated with difference number of feeders (namely 3, 5, and 10 feeders), and other system parameters such as limitation on carrying capacity, working time, traveling time of the mobile robot, planning horizon, and characteristics of feeders. The robot is designed to carry up to 3 SLCs at a time to perform part-feeding tasks during a given planning horizon of one hour (corresponding to an eighth of a full production shift). The maximum and minimum levels of parts of feeders are respectively distributed within the ranges of [300, 2000] and [100, 1000] while part-feeding rates in seconds are in-between the interval [1.5, 4.5]. The working times of the robots in seconds at feeders and the warehouse per SLC are respectively distributed within the range of [40, 60] and [25, 40] while the traveling times of the robot in seconds are in-between the interval [20, 60]. Note that the cost matrix of the generated traveling times should satisfy the triangle inequality.

For GA parameters, the population size, P_c , P_m , G_m , and G_c are set to be 100, 0.6, 0.2, 500, and 100, respectively. The weighted-sum fitness function F (Equation 3) will be calculated using one of three different values of the weight coefficient α , namely, 0.2, 0.5, and 0.8. The proposed heuristic has been coded in VB.NET, and all the problem instances run on a PC having an Intel® Core i5 2.67 GHz processor and 4 GB RAM. The results for three randomly generated problems in combination with three values of the weighted coefficient α are presented in Table 1 below.

Table 1. The best solutions of three generated problems

Problem	No. of feeder	No. of subtasks of tasks	Weighted coefficient (α)	Total traveling time of robot (second)	Total tardiness of tasks (second)	Computation time (second)
1	3	11	0.2	432	0	0,49
			0.5	432	0	0,63
			0.8	428	6	0,41
2	5	24	0.2	706	0	1,00
			0.5	690	4	1,12
			0.8	682	15	1,10

3	10	42	0.2	1564	0	2,52
			0.5	1542	22	2,70
			0.8	1528	55	2,40

The total traveling time of the robot, total tardiness of tasks, and computation time shown in Table 1 are the average of 10 runs. It can be observed that as the weighted coefficient α increases, two objectives of each problem instance have opposite trends where the total traveling time of the robot decreases and the total tardiness of tasks increases. In other words, as saving battery energy allowing the robot to be utilized in a longer duration is more important, the robot has a tendency to travel less and vice versa. Similar explanation is also applicable to the total tardiness of part-feeding tasks. Such kinds of solutions in Table 1 are non-dominated solutions for which no improvement in any objective function is possible without sacrificing the other objective function. It also shows that when the size of the problems grows, the computation time of the GA-based heuristic becomes longer, but it is still acceptable (i.e., the largest problem size with 10 feeders and the coefficient α of 0.5 requires 2.7 seconds in average to find the efficient solution). These results provide evidence to prove that the GA-based heuristics could be used to produce efficient schedules within reasonable time in online production mode.

The above solutions are initial schedules for the robot. These schedules serve as input to a Mission Planner and Control (MPC) program which is accessed by using XML-based TCP/IP communication to interact with the robot, Manufacturing Execution System (MES), and the module of GA-based heuristic. In practice, there might be some errors in manufacturing such as machine breakdown, or changes in manufacturing conditions such as characteristics of feeders (e.g. minimum levels of parts), or carrying capacity of the robot. These events will be reported by the MES so that the MPC program can update current states of the shop floor and then call the heuristic module to reschedule part-feeding tasks of the robot. By relaxing the last assumption mentioned in Section 2, the proposed heuristic in turn will use the current states as new input, re-optimize to get alternative schedules, and send these schedules back to the MPC program.

5 Conclusions

In this paper, a problem of scheduling a single mobile robot to carry out part-feeding tasks of production lines is studied. To maintain the manufacture of products, it is important for planners to determine feeding sequences which minimize the total traveling time of the robot and the total tardiness of the feeding tasks while taking into account a number of practical constraints. The main novelty of this research lies in the consideration of dynamic time windows and limitation on carrying capacity where multiple routes have to be performed by the single mobile robot. A genetic algorithm-based heuristic was proposed to find efficient solutions for the problem. The results in the numerical examples showed that the proposed heuristic is fast enough to be used to generate efficient schedules compromising the objectives in online production mode. The heuristic may be also used to produce alternative schedules in rescheduling

scenarios when there might be some errors or changes in manufacturing conditions. Moreover, the heuristic could be considered to deal with more performance criteria according to requirements of planners, and by investigating different scenarios with various weighted coefficients of those criteria, it can specify which schemes are more beneficial for the manufacturing. For further research, a general model of scheduling multiple mobile robots should be considered together with rescheduling mechanisms to deal with real-time disturbances.

Acknowledgments This work has partly been supported by the European Commission under grant agreement number FP7-260026-TAPAS.

References

1. Bocewicz, G., Bach, I., Wójcik, R.: Production Flow Prototyping subject to Imprecise Activity Specification. *Kybernetes* 38, 1298–1316 (2009)
2. Choi, I.C., Kim, S.I., Kim, H.S.: A Genetic Algorithm with a Mixed Region Search for the Asymmetric Traveling Salesman Problem. *Comput. Oper. Res.* 30, 773–786 (2003)
3. Crama, Y., Kats, V., van de Klundert, J., Levner, E.: Cyclic Scheduling in Robotic Flowshops. *Ann. Oper. Res.* 96, 97–124 (2000)
4. Dang, Q.V., Nielsen, I., Bocewicz, G.: A Genetic Algorithm-based Heuristic for Part-feeding Mobile Robot Scheduling Problem. In: Rodríguez, J.M.C. (eds.) *Trends in PAAMS. AISC*, vol. 157, pp. 85–92. Springer-Verlag, Heidelberg (2012)
5. Dang, Q.V., Nielsen, I., Steger-Jensen, K., Madsen, O.: Scheduling a Single Mobile Robot for Part-feeding Tasks of Production Lines. *J. Intell. Manuf.* (2012) (accepted)
6. Gen, M., Lin, L.: *Network Models and Optimization*. Springer-Verlag, London (2008)
7. Gerns, R., Goldengorin, B., Turkensteen, M.: Lower Tolerance-Based Branch and Bound Algorithms for the ATSP. *Comput. Oper. Res.* 39, 291–298 (2012)
8. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, New York (1989)
9. Hurink, J., Knust, S.: A Tabu Search Algorithm for Scheduling a Single Robot in a Job-shop Environment. *Discrete Appl. Math.* 119, 181–203 (2002)
10. Liu, F., Zeng, G.: Study of Genetic Algorithm with Reinforcement Learning to Solve the TSP. *Expert Syst. Appl.* 36, 6995–7001 (2009)
11. Maimon, O., Braha, D., Seth, V.: A Neural Network Approach for a Robot Task Sequencing Problem. *Artif. Intell. Eng.* 14, 175–189 (2000)
12. Moon, C., Kim, J., Choi, G., Seo, Y.: An Efficient Genetic Algorithm for the Traveling Salesman Problem with Precedence Constraints. *Eur. J. Oper. Res.* 140, 606–617 (2002)
13. Snyder, L.V., Daskin, M.S.: A Random-Key Genetic Algorithm for the Generalized Traveling Salesman Problem. *Eur. J. Oper. Res.* 174, 38–53 (2006)
14. Xing, L.N., Chen, Y.W., Yang, K.W., Hou, F., et al.: A Hybrid Approach Combining an Improved Genetic Algorithm and Optimization Strategies for the Asymmetric Traveling Salesman Problem. *Eng. Appl. Artif. Intell.* 21, 1370–1380 (2008)
15. Zacharia, P.Th., Aspragathos, N.A.: Optimal Robot Task Scheduling Based on Genetic Algorithms. *Robot Comput. Integrated Manuf.* 21, 67–79 (2005)