Science Arts & Métiers (SAM)
is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: https://sam.ensam.eu
Handle ID: .http://hdl.handle.net/10985/13415

To cite this version :

Cristóvão SILVA, Nathalie KLEMENT, Olivier GIBARU - A generic decision support tool to planning and assignment problems: Industrial application & Industry 4.0 - In: 27th International Conference on Flexible Automation and Intelligent Manufacturing, Italie, 2017-06 - Procedia Manufacturing - 2017

Any correspondence concerning this service should be sent to the repository Administrator : scienceouverte@ensam.eu

Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 00 (2017) 000–000

This copy has been created with permission of Elsevier and is reserved to FAIM2017 conference authors and participants.

27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy

A generic decision support tool to planning and assignment problems: Industrial application & Industry 4.0

Nathalie Klement*[a], Cristóvão Silva[b], Olivier Gibaru[a]

[a]*LSIS CNR UMR 7296 (équipe INSM). École Nationale Supérieur des Arts et Métiers. 8, boulevard Louis XIV – 59046 Lille Cedex, France*
[b] *CEMUC, Department of Mechanical Engineering, University of Coimbra, Rua Luís Reis Santos, 3030-788, Coimbra, Portugal*

Abstract

Decision support tools are essential to help the management of industrial systems at different levels: strategic to size the system; tactical to plan activities or assign resources; operational to schedule activities. We present a generic and modular decision support tool to solve different problems of planning, assignment, scheduling or lot-sizing. Our tool uses a hybridization between a metaheuristic and a list algorithm. The specification of the considered problem is considered into the list algorithm. Several tactical and operational problems have been solved with our tool: a problem of planning activities with resources assignment for hospital systems, a lot-sizing and scheduling problem taking into account the setup time for plastic injection, and a scheduling problem with precedence constraints. At the strategic level, this tool can also be used as part of the Industry 4.0 to design reconfigurable production systems. This paper summarizes some problems solved with the proposed tool, and presents the evolution of our tool.

© 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 27th International Conference on Flexible Automation and Intelligent Manufacturing.

Keywords: Agile manufacturing; Scheduling algorithms; Heuristic searches; Resource allocation; Industrial production system.

* Corresponding author. Tel.: +33 3 20 62 27 69.
E-mail address: nathalie.klement@ensam.eu

2351-9789 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 27th International Conference on Flexible Automation and Intelligent Manufacturing.

1. Introduction

Industry 4.0 is the main international program which aims at improving the operational system in companies. More
companies are concerned by this approach. Thanks to Internet of things, "things" are connected to ease the communication, but other improvements may be envisaged. An integration of the whole production system is needed. Managers need to fully control the production, the actual one and the future one. Decision Support Systems are needed to help managers to decide. Because of Industry 4.0, more data are available, more variability can be considered, so DSS become necessary. How can the future demand be integrated without major changes in the actual layout? The production system needs to be flexible to face the variety of products and the quantities of products. A lot of problematics are concerned by this 4th industrial revolution: sizing of the shop floor, sizing of resources, planning of activities, assignment of resources, scheduling of activities. To propose some decision support tools to help the manager, new trends need to be considered such as continuous improvement, Big Data or collaborative robotics [1]. For instance, it would be necessary to use the data in the shop floor to treat them in real time to adapt the schedule and the planning to the hazards, a link with the used ERP by the company needs to be done. Thanks to collaborative robotics, flexible production means can be used in our future flexible and agile production system, which will be reconfigurable. Many companies are actually thinking about converting their actual system into a reconfigurable production system.

We propose a decision support tool which can be used to design reconfigurable production system. At a beginning, three static problems have been studied, in which the demands are already known. We focused on many problems: planning, scheduling, resources assignment. These problems can be summarized: they represent a system in which activities have to be done over a horizon planning. Each activity has some characteristics such as processing time and needed resources. Some resources are available to process the activities. The system is ruled by some constraints. Different objectives can be achieved such as optimizing the productivity of the system. Once this tool has been validated for the static problems, we can focus on the dynamics ones: the demands are not completely known, they can vary in quantity and/or variety of products.

Section 2 presents the generic and modular decision support tool we developed. Some examples of problems that have been identified and solved are presented in Section 3. Our future work will focus on the reconfigurable production systems, presented in Section 4. This paper ends by a conclusion in Section 5.

2. Generic and modular decision support tool

2.1. Genericity

The proposed tool, illustrated by Fig. 1, uses a hybridization of a metaheuristic and a heuristic, specifically a list algorithm. A single solution based metaheuristic or a population based metaheuristic can be used. The encoding used by the metaheuristic is a list Y of activities. List algorithm L considers the activities according to their order in list Y, to plan and assign them to the required resources, considering the problem constraints. This builds solution X. Objective function H evaluates solution X. According to this evaluation, the solution is chosen or not by the metaheuristic. At the end of the computation, the given solution by the hybridization is the best list Y^* of activities: the one which optimizes the objective function by applying the list algorithm. This hybridization can be used to solve many problems: only the list algorithm and the objective function need to be specific to the considered problem by integrating the different constraints which rule the system, and the objectives to achieve.

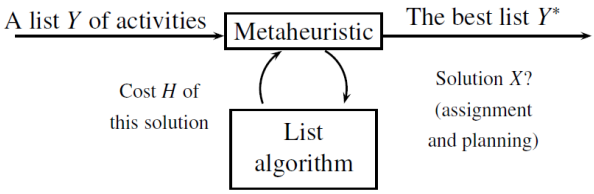

Fig. 1. Hybridization Metaheuristic – List algorithm

2.2. A list Y of activities

The general scheme of the encoding is given by Equation (1), with Ω the set of all lists Y and S the set of all admissible solutions X built by the list algorithm L. Ω is the set of all permutations of activities. Cardinal of Ω is N! with N the number of activities. Y ∈ Ω is a list of activities. More details about the encoding are given in [2].

$$Y \in \Omega \xrightarrow{Heuristic\ L} L(X) = X \in S \xrightarrow{Criterion\ H} H(X) \tag{1}$$

2.3. Metaheuristic

The metaheuristic performs in Ω. An initial solution is randomly computed: a list of activities randomly sorted between one and the number of activities. A neighborhood system is used to visit the set of solutions; it allows to go from one solution to another one. Neighborhood system V is a permutation of two activities in list Y: the activity at position i permutes with the one at position j, with i and j two different random numbers. V satisfies the accessibility and reversibility properties. Several metaheuristics have been used: some single solution based metaheuristics such as iterated local search and simulated annealing.

A simple descent stays in the first found local minimum. [3] showed that an iterated local search allows to go out from this local minimum. After having applied a local search, the current solution is disrupted to go out from the local minimum. A new local search is applied from the disrupted solution. Kangaroo algorithm is an iterated local search. It consists in applying a stochastic descent, but if there is no improvement of the current solution during A iterations, a jump is made. The used formula to compute the number of iterations A is given by Equations (2) and (3). To make this jump, a solution is chosen in a neighborhood system W, different from V. Kangaroo algorithm converges in probability to the set of optimal solutions if neighborhood system W satisfies the accessibility property. We choose W as the consecutive application five times of V.

$$A \geq card(V) * \ln(2) \tag{2}$$

$$card(V) = \frac{N*(N-1)}{2} \tag{3}$$

Simulated annealing is inspired by a process used in metallurgy which consists of alternating cycles of slow cooling and heating. Inhomogeneous simulated annealing was used by [4] to simulate the physical cooling in metallurgy. Applied to the optimization field, it consists in executing a descent with a non-zero probability to choose a worst solution than the current one. This probability decreases while the number of iterations increases. [5] proved that simulated annealing converges in probability to the set of optimal solutions if neighborhood system V satisfies the accessibility and reversibility properties. Two parameters are used. The initial temperature T_0 is chosen such as all the transitions are accepted at the beginning, defined by Equation (4). The decreasing factor α is chosen such as the final temperature T_a is close to zero, computed as Equation (5), with IterMax the maximum number of iterations.

$$e^{-\frac{H(X')-H(X)}{T_0}} \approx 1, \forall(X, X') \tag{4}$$

$$\alpha = \sqrt[IterMax]{\frac{T_a}{T_0}} \tag{5}$$

2.4. List algorithm

A list algorithm is used to build the solution X from the list Y: it assigns the activities to resources over the horizon planning according to the problem constraints. List scheduling algorithms are one-pass heuristics that are widely used to make schedules. [6] defined standard list scheduling algorithm as the construction of a schedule by assigning each activity in listed order to the first resource that becomes idle. It is important to work with a list algorithm because the metaheuristic browses the set of lists Y. So the used algorithm needs to consider the order of

the list to assign activities to resources over the horizon planning. The developed list algorithms are detailed in Section 3.

2.5. Objective function

Solutions are compared according to an objective function which characterizes the quality of the solution. The aim of our tool is to find the solution X^* defined by Equation (6). The used objective functions are detailed in Section 3. The weighed criteria method defined by [7] is used. The objective function is a weighed sum between some criteria. An example with two criteria $H_1$and H_2 is defined by Equation (7). ω_2 is defined by Equation (8), so both criteria can be easily readable. This method can be used with more than two criteria.

$$X^* = \min_{\forall X \in S} H(X) \quad (6)$$

$$H(X) = 10^{\omega_2} * H_2(X) + H_1(X) \quad (7)$$

$$10^{\omega_2} > \max_{\forall X \in S} H_1(X) \quad (8)$$

2.6. The best list $Y*$

Algorithm 1 describes the whole method with the example of simulated annealing as the used metaheuristic. Set Ω of lists Y of activities is browsed thanks to the metaheuristic using neighbourhood system V. Lists are compared thanks to list algorithm L and objective function H. According to an acceptance criterion, some lists are selected. At the end, the metaheuristic gives the best found list Y^*.

Algorithm 1: Hybridization of simulated annealing and a list algorithm

```
Data: Initial solution Y ; Temperature T_0 ; Decreasing factor α
T := T_0 ; X := L(Y)
while necessary do
    Choose uniformly and randomly Y' ∈ V(Y)
    X' := L(Y')
    if H(X') < H(X*) then
        X* := X' ; Y* := Y'
    if H(X') ≤ H(X) then
        X := X' ; Y := Y'
    else if rand[0, 1) ≤ e^(-(H(X')-H(X))/T) then
        X := X' ; Y := Y'
    Compute the new temperature T := α × T
return Y*
```

3. Applications

In the industrial engineering field, many problems can be solved according to the used decision level. Some of these problems are summarized in Table 1. The applications presented in the current section are about tactical or operational level. Data used as an entrance of our tool and results given by our tool are different according to the considered level. For example, at the operational level, production plan is a result given the description of the shop floor, while at the strategic level, description of the shop floor is a result given the production plan. Each problem is detailed in the same way: analysis, the two specific parts (list algorithm and objective function), some results.

Table 1. Decision levels and problems.

Decision level	Horizon timing	Studied problem
Strategic	Years	Sizing of the problem
Tactical	Months	Activities planning & Resources assignment
Operational	Weeks, Days	Activities scheduling & Response to hazards

3.1. Activities planning and resources assignment

Our method has firstly been used to solve an activities planning and resources assignment problem in a multi-place hospital context [8]. The application has been made for the medical imaging department. The planning horizon is divided into periods. Each period represents one half-day. Activities are exams. The hospital system is composed by several places. Some material resources belong to each place. Each material resource has an opening schedule. All material resources cannot treat all the exams, there are some incompatibilities between exams and material resources. Each exam has a given processing time and a due date before it has to be done. The objective is to assign each exam to one material resource during one period, respecting the time and compatibility constraints.

This problem has been analyzed as a bin packing problem [9]. The proposed list algorithm assigns each exam to one resource and one period. It is an extension of an existing list algorithm to solve the bin packing problem: First Fit. It consists in assigning the exams to the first available couple (resource, period). Compatibility between exam and couple must be respected. Exams have to be done as soon as possible so all resources from one period are tested before going to the next period. Current exam Y_i is assigned to couple (resource, period) if available time in this couple is bigger than the processing time of exam Y_i and if exam Y_i is compatible with the resource. A new period is considered if there is no couple in the current one with enough time to receive the exam.

The objective function represents the timing aspect of our problem. Exams have to be done as soon as possible, thus the makespan, the period assigned to the last exam, should be considered. Because many solutions may have the same makespan, we choose instead the sum of assigned periods to all exams, so the solutions can be dissociated. Another criterion is considered to ensure that most of the exams are assigned before their due date. This criterion is computed as the number of exams assigned after their due date.

Our tool provides a good planning of exams, with the assignment of one resource and one period to each exam, minimizing the objective function. We could also add the assignment to human resources. Human resources can move over the different places belonging to the hospital system. Each human resource has a planning defining its availability. Each human resource has some skills to use some material resource. The new objective could be to assign each exam to one material resource and one human resource during one period. Some additional constraints have to be considered for each assigned exam: the capability of the human resource to work on the material resource, the availability and the location of the human resource at the place where is the material resource.

3.2. Lot-sizing and scheduling

The second solved problem with our tool is a lot-sizing and scheduling problem with setup and due dates, for the injection plastic case [10]. Activities are jobs to schedule. A set of N jobs has to be scheduled on shared machines with their respective mold. Each job has a given processing time and an associated due date. A sequence dependent setup time is required when the production changes over from a job requiring a given mold to a job requiring a different one. A job is not allowed to be split but several jobs, requiring the same mold, may be grouped together to form one lot and, thus, saving setup costs. Due to compatibility factors, each mold can only be allocated to a subset of the available machines. Each mold is unique; thus the same mold cannot be allocated to different machines during the same time period. The objective is to allocate jobs to each available machine and define the processing sequence on each machine in order to minimize the total tardiness.

The proposed list algorithm to solve this problem is given by Algorithm 2.

The evaluation of a solution is made according to the value of the total tardiness. For a given solution, for each job, the difference between its due date and its actual final date given by the solution is computed. Then the sum of all tardiness for all jobs is made. The number of setups is not considered because all that matters is to minimize the tardiness while delivering the jobs.

For small instances (2 machines, 10 or 15 jobs), the solutions built by our method were compared to the optimal one obtained by solving a mathematical model with CPLEX. It was found that our proposed method gives an optimal solution for these two instances. Next, our solutions were compared to the ones daily used by the injection company. Thanks to our method, an average reduction of 25% of tardiness is achieved.

Algorithm 2: List Algorithm for an injection problem

```
Data: List of jobs (Y_i)_{i∈{1,N}}, Processing time of all jobs
forall the i do
    Order the machines according to their release date
    First machine
    while Job Y_i is not assigned do
        if Job Y_i and the machine are compatible then
            if The needed mold is available then
                if The actual used mold on the machine is the good one then
                    Actualize the release date of the machine without setup
                else
                    Actualize the release date of the machine with setup
                Assign the job
            else
                Assign the job to the machine which uses the needed mold
            Actualize the release date of that machine, taking into account the setup if needed
        Next machine
```

3.3. Scheduling with precedence constraints

The third application is a scheduling problem considering precedence constraints between activities. A set of metallic components is to be produced to satisfy the demand from an assembly line. Each component has to follow a processing sequence to be produced and each operation in this sequence requires a given machine. The planning horizon is a week which is divided in five periods of one day. To satisfy the demand from the assembly line, a set of different lots of components is to be produced in each day of the planning horizon. Thus, a set of N jobs has to be processed on a set of M machines. Each job is defined by a sequence of operations that are associated with a particular machine. Each operation has a processing time and there is a setup time between the processing of two consecutive operations which is sequence dependent. Each job has a requested period. Each machine has an opening schedule. We consider a penalty function, composed by two parts: a storage cost (earliness) if the job is produced in a period prior to the requested one, and a tardiness cost if the job is produced in a period after the requested one. The objective is to define the operations sequence in each machine in order to minimize the total penalty.

The list algorithm works in two phases. During the first one, operations are assigned to the needed machine, respecting the precedence constraints, during the requested period. The second phase revises the solution previously built: if operations are assigned overtime, they are moved to the previous or next period, respecting the precedence constraints. If all jobs cannot be done during that week, they will be strongly penalized and done next week.

The evaluation of a solution is made according to the value of the total lateness as defined by the penalty function. For each job, we compute the difference between the period in which it is concluded and the period for which it has been required. The weights of earliness and tardiness are different.

One instance, representative of the reality, has been created. Result of using our method is promising. It was decided that the researchers involved in this project will continue their collaboration with the company in order to keep developing the proposed tool to solve real size problems.

4. Industry 4.0

At operational and tactical levels, the previous examples may be part of the context of Industry 4.0. Thanks to the Information System, results given by our tool considering planning, scheduling, assignment, and lot-sizing problems can be transmitted live to the shop floor. For instance, the shop floor can apply the determined schedule and assignment by producing the right product with the right resource at the right time. Dynamics problems could also

be considered, by defining an appropriate horizon planning. If hazards or variability of the demands are considered, the running of our tool may be done often, to actualize the schedule and assignment in response to these changes.

At a strategic level, we intend to size systems for the next years. One objective is to determine the needed number of resources. But because the variety and quantity of products are not stable and not easily predictable for the next years, we should consider new trends, defined by the Industry 4.0 project.

4.1. Agility and flexibility

The current market context is characterized by global competition between industries, high product variety and variable volumes. Those are keys that require the launch of products with a short life cycle and a high degree of customization. To do so, reconfigurable layout has been introduced. One first definition has been given by [11] as "the change of an existing plant configuration to another that optimizes costs and time". Since then, many researchers are working on the subject. [12] summarized in Fig. 2 different strategies of reconfigurability.

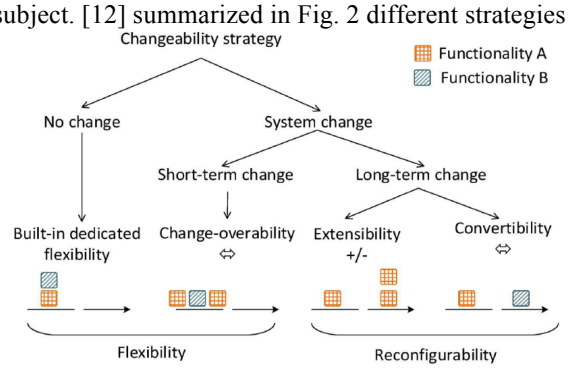

Fig. 2. Changeability strategies

Our proposed tool may be used to solve such problems, where production system should be convertible to face the increasing product variety, and extensible to face the increasing volumes. The method first starts with an analysis of the system: what are the demands or future demands and what are the constraints ruling the system. At the opposite of the previous applications, type and number of resources are a result given by our tool, according to the production. Many factories or companies are actually focusing on this problematic. To help them to understand better our proposed solution, we could also use some simulation tool. The optimization part, using the proposed tool, will define one good solution of the problem. This solution is then simulated, using the data or future data of the company, with performance indicators that the company is used to understand. Fig. 3 summarizes the method.

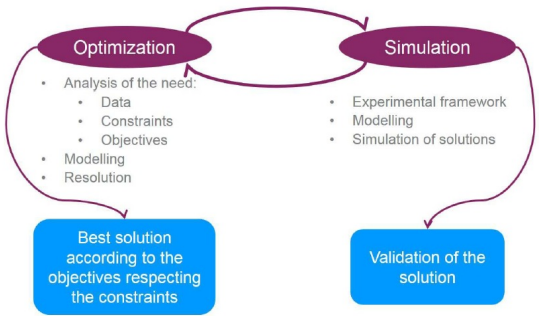

Fig. 3. Hybridization: optimization and simulation

4.2. Collaborative robotics

Thanks to collaborative robotics, robots and human can work together. Safety barriers are not needed anymore. Security has increased because these robots are more sensible. Thanks to many captors, they know when the human is close to them. So they can adapt their work and speed. If a contact is made between the robot and the human, the robot stops and needs a human action to start again its work. These robots are less imposing than traditional robots.

Because safety barriers are not needed, these robots can easily be moved from one place to another place of the layout. Mobile collaborative robotics is also existing, where the collaborative robot is put on a mobile platform. In this way, the robot can move independently. Moreover, these robots are easily configurable. While with a traditional robot or machine, a changing of production needs a changing of tools with setup, a collaborative robot only needs to change its program. Collaborative robotics should be used in reconfigurable production system. They will be compatible with more products, they need less or shortest setups, they can move from one place to another in the layout and they are smaller than the traditional machines. All these characteristics can easily be modeled and better used with our tool.

4.3. Big Data

Some new technologies have emerged the last years: Internet of Things, RFID, Cloud Computing, Mobile Devices, etc. Each company can have much more information about the running of its system: it is called Big Data. Manufacturing Execution Systems exploits Big Data to collect data about the system in real time. [13] discussed about the future of MES. It has to be decentralized, vertical integrated to consider the whole supply chain, mobile and connected, and using the cloud computing. The future evolution of our tool needs to consider Big Data. But how to collect every available data? How to treat these data? And how to exploit them? Big Data can feed our tool to generate more entry data to describe the system. The better the system is known, the better the proposed solution will be. Once our tool has built some solution, it can feed the MES. For example, if the proposed tool is used at an operational level, it will build the schedule of activities day to day. This schedule is transferred to the required resource. Another use is the response to hazards: if some resources are missing, our tool needs to be efficient enough to compute in real time a new schedule considering the missing resources. Big Data, which is an axis of study part of Industry 4.0, can improve the use of our tool.

5. Conclusion

Because of the quick evolution of the market, future demand in finished products is highly not predictable in quantity and variety. Production system needs to be adapted. To do so, some new definitions of production systems have been given: they need to be reconfigurable, agile, flexible or changeable. But how to design them and maintain them? We proposed a generic and modular decision support tool. It already solved many problems with a static dimension where the demand is known. It can now be used to consider reconfigurable production system. It can design a new layout, defining the right number of needed resources, considering the future demand. Because this demand is supposed highly variable, some new means of production such as collaborative robotics main be considered. Many information is needed to describe the system. Big Data needs to be integrated to our tool. Once the system has been designed, our tool could be daily used, to schedule the activities, facing the possible hazards. Our method can build a good solution in less than a few minutes. Our research activity is in the heart of industrial problematic in the context of Industry 4.0. Many companies are thinking about new systems which can face the future demand in variety and/or product. This purpose is experimented in our laboratory with an experimental field.

References

[1] J. Guérin, O. Gibaru, E. Nyiri, S. Thiery. IECON Industrial Electronics Conference, 2016.
[2] M. Gourgand, N. Grangeon, N. Klement, Advances in Production Management Systems, volume 438, (2014) 572–579.
[3] H. R. Lourenço, O. Martin, T. Stutzle, F. Glover, G. Kochenberger, Handbook of Metaheuristics (2002) 321–353.
[4] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, The Journal of Chemical Physics 21 (1953) 1087–1092.
[5] E. Aarts, P. van Laarhoven, Simulated Annealing: Theory and Applications, Kluwer Academic Publishers, 1987.
[6] X. Zhu, W. E. Wilhelm, IIE transactions 38 (2006) 987–1007.
[7] C. A. Coello, ACM Computing Surveys (CSUR) 32 (2000) 109–143.
[8] N. Klement, M. Gourgand, N. Grangeon. 10th International Conference on Health Informatics, 2017.
[9] M. Gourgand, N. Grangeon, N. Klement, 7th International Conference on Health Informatics, Angers, France, pp. 117–124.
[10] C. Silva, N. Klement, O. Gibaru, International Joint Conference - CIO-ICIEOM-IIE-AIM (IJC 2016), San Sebastian, Spain, ICIEOM.
[11] T. A. Lacksonen, C.-Y. Hung, IIE transactions 30 (1997) 91–99.
[12] N. Benkamoun, K. Kouiss, A.-L. Huyet, Organisation and Management, Milan, Italy, 27-30.07. 15.
[13] F. Almada-Lobo, Journal of Innovation Management 3 (2016) 16–21.