A GRASPxELS for Scheduling of Job-Shop Like Manufacturing Systems and CO2 Emission Reduction

Sylverin Kemmoe Tchomte, Nikolay Tchernev

A GRASPxELS for scheduling of job-shop like manufacturing systems and CO_2 emission reduction

Sylverin Kemmoe Tchomte[1] and Nikolay Tchernev[2]

[1]CRCGM EA 3849, Université d'Auvergne, Clermont Ferrand, France
Sylverin.Kemmoe_Tchomte@udamail.fr
[2]LIMOS UMR CNRS 6158, Université d'Auvergne, Aubière, France
Nikolay.Tchernev@udamail.fr

Abstract. The issue of reducing CO_2 emission and associated carbon footprint consumption for manufacturing scheduling is addressed. We focus our attention on a job-shop environment where machines can work at different speeds and therefore different energies consumed, i.e. CO_2 emissions. It represents an extension of the classical job-shop scheduling problem, where each operation has to be executed by one machine and this machine can work at different speeds, problem which has been introduced by [1]. Energy-efficient scheduling of such type of manufacturing systems demands an optimization approach whose dual objectives are to minimize both the CO_2 emissions and the makespan. To solve this new problem, a GRASPxELS is developed. New instances benchmark based on well know Laurence's instances are introduced and numerical experiments are proposed trying to evaluate the method convergence. The performance is evaluated using the optimal solutions found after a strongly time consuming resolution based on a linear formulation of the problem.

Keywords: energy efficiency, job-shop, scheduling, CO_2 emissions, GRASP

1 Introduction.

Nowadays, companies are not only facing economics trends with objective to improve profitability and competitiveness, but also reducing their CO_2 emissions using the green manufacturing concepts. In this paper, we focus our attention in a job-shop like manufacturing environment where machines can work at different speeds and therefore different energies consumed, i.e. different CO_2 emissions generated. When machine speed is high the processing time of the job operation is short and the energy consumption CO_2 emission is high. Contrary if the speed is low, the processing time increases and the energy consumption decreases and CO_2 emissions are lower. In this type of manufacturing systems there is a close relation between lead times and CO_2 emissions. To this end, we analyse the relationship between machine speed, makespan and CO_2 emissions in order to obtain a multi-objective solution. Our goal is to find a solution that minimizes hierarchically the makespan and the energy consumption, and therefore CO_2 emissions. To model the problem we chose the Job Shop with different

Speed Machine theoretical problem (*JSSM*) first formulated by [1] which problem is an extension of the classical *JS* problem, where each job operation should be processed at a determined speed. This paper presents a *GRASPxELS* metaheuristic approach for the *JSSM* problem solving that considers CO_2 emissions and associated carbon footprint in addition to makespan.

This paper calls for the development of more specialized algorithms for this new scheduling problem and examines computationally tractable approaches for finding near-optimal schedules. The rest of the paper is organized as follows. In section 2 the problem is presented. In section 3 a *GRASPxELS* with productivity and environmental objectives is presented. Section 4 is dedicated to the numerical experiments, before some concluding remarks.

2 Job Shop with Different Speed Machine

2.1 Problem description

Formally the job-shop scheduling problem with different speed machine (*JSSM*) can be defined as follows. The *JSSM* consists of a finite set J of n jobs $\{J_i\}_{i=1}^{n}$ to be processed on a finite set M of m machines $\{M_k\}_{k=1}^{m}$. Each job J_i consists of a sequence of m_i operations $O_{i,1}, O_{i,2}, O_{i,3} \ldots, O_{i,m_i}$. Each operation $O_{i,j}$ is associated with a particular job i and machine j and has an integer duration $p_{O_{i,j}}^{\mathrm{mod}}$ and generate CO_2 emissions depending on the speed v. Each machine can work with different speeds, each speed depends on the processing mode and is linked up to a duration and CO_2 emissions. There are $N = \sum_{i=1}^{n} m_i$ operations in total and therefore, the dimension of the problem is often denoted as $n \times m \times |NbSpeed|$. A feasible solution is a complete definition of operation starting times that satisfies the following constraints: (*i*) delivery times of the products are undefined; (*ii*) no more than one operation of any job can be executed simultaneously; (*iii*) no machine can process more than one operation at the same time; (*iv*) the job operations must be executed in a predefined sequence and mode and once an operation is started, no pre-emption is permitted. The objective is to find a feasible schedule that minimizes the completion time of all the tasks and the energy used, i.e. CO_2 emissions. According to the $\alpha|\beta|\gamma$ notation introduced by [2] the problem can be represented by $J| \, |C_{\max}$ and is known to be NP-hard [2]. In addition an association between duration and energy has been created. For each job operation three different speeds have been defined. Each speed has its own duration and CO_2 emissions generated depending on the energy consumption. When the work speed increases the energy and CO_2 emissions also increase and on the other hand the duration decreases.

2.2 Related Literature

Although the *JS* is well-addressed in the literature (see for survey [3] and some recent papers [4, 5], from best of our knowledge only few articles [1], [6, 7] are concerned with energy consumption. In [6] bi-objective model for energy consumption and makespan optimization for *JS* is formulated and heuristic algorithm is developed to locate the optimal or near optimal solutions of the model based on the Tabu search mechanism. [7] proposed a mixed-integer nonlinear programming model for the hybrid flow shop scheduling problem to minimize energy consumption. An improved genetic algorithm solved this efficiently. Although energy consumption was mainly considered and the makespan was a key constraint, they ignored on-peak times for energy use.

[1] is the first paper which considers the energy in JSSM. In this seminal paper, using constraint programming technology for minimizing makespan and energy consumption, the authors analyzed energy-efficiency, robustness and makespan, and the relationship among them.

In the specialized literature about production scheduling research on minimizing the energy consumption of manufacturing systems has focused on following shop scheduling problems: (*i*) single machine [8]; (*ii*) flow-shop [9]; (*iii*) hybrid and flexible flow-shop [10, 11].

Although many researchers have addressed energy consumption in scheduling, one of them have tried to consider *JSSM* problem. None of this research has proposed a metaheuristic approach for *JSSM* problem.

3 GRASPxELS Based Approach for *JSSM* Solving

In order to find approximate solutions to the *JSSM*, we propose a *GRASPxELS* metaheuristic approach base on Greedy Randomized Adaptive Search Procedure (GRASP) hybridized with an evolutionary local search (*ELS*). Proposed by [12], *GRASPxELS* approach has been used successfully and competitive results have been reported in the literature for the classical *JS* problem [5]. The purpose of this section is to evoke the principles of *GRASP×ELS* where:

- *GRASP* is a multi-start local search metaheuristic in which each initial solution is constructed using a greedy randomized heuristic. The multi-start approach of the *GRASP* provides *np*>1 initial solutions, improved by a local search. It was first presented in [13, 14], by Feo and Resende, and later formalized and given its acronym in [15]. Since then, it has been used to solve a wide range of problems with many and varied applications in the real life such as the design of communication networks, scheduling, collection and delivery operations and computational biology. For recent and comprehensive surveys of *GRASP* we refer the reader to [16, 17, 18].
- The purpose of *ELS* is to better investigate the current local optimum neighborhood during *ni* iterations, before leaving it. Starting from an initial solution, each *ELS* iteration consists in taking a *ns* >1 copies of the incumbent solution *S*, applying a

mutation (child solution) and improving the mutated solutions using a local search. The resulting best solution *S** becomes the incumbent solution *S*.

The proposed *GRASPxELS* is based on the following key features which enable scheduling problems solving using evolutionary algorithms and favours efficient global process for solution space investigation:

- Graph representation such as disjunctive/conjunctive graph [19];
- A Quasi-Direct Representation of Solution that is not a whole solution of the problem but a compact representation, such as a sequence of nodes or operations. Bierwirth in 1995 [20] introduces an alternative representation as a sequence of job number. This kind of representation is called: sequence with repetition. Based on his proposal, the solution of Fig. 1 is encoded to: 1 2 3 3 2 3 2 1 1.
- An efficient local search taking advantages of the longest path analysis using, well-known the neighbourhood system of [21] which concern two consecutive operations on the longest path.

Algorithm 1 illustrates the *GRASPxELS* principles and implementation in psedo-code. The *GRASPxELS* iterations are carried out in line 1-18. In line 1, the variable that stores the best solution found is initialized. The block of instructions between lines 2 and 18 is executed iteratively, where each iteration consists of three phases:

- Construction phase (line 3): initial solutions are built, one element at a time, with a greedy randomized heuristic. At each construction iteration the next element to be added is determined by ordering all elements in a candidate list. The probabilistic component of a *GRASPxELS* in this phase is characterized by randomly choosing one of the candidates in the list, not always is the top best.
- Local search phase (lines 4-6): since this solution of the construction phase is not guaranteed to be locally optimal, a local search is performed to minimize the makespan. In line 5, for the last found minimal makespan a local search for minimizing the CO_2 emissions is performed. The quality of the obtained solution is compared to the current best found and, if necessary, the solution is updated (line 6). The use of local search phase based on two search procedures permits to order by strict preference makespan and CO_2 objectives, where the first one is the most preferred. The makespan is the most preferred since it correspond on due dates of customer orders. Using the first level optimization makespan is minimized (see solution Fig2.). The next level minimization the CO_2 emissions are minimized by increasing some operation durations for example J2, J3 and J5 and consequently reduce CO_2 emissions as shown in Fig. 3.
- Evolutionary local search (*ELS*) phase (lines 8-17): to better investigate the current local optimum neighborhood. In line 16, the quality of the obtained solution is compared to the current best found and, if necessary, the solution is updated.

The successive changes of encoding occur between the local search and mutations steps of our *GRASPxELS*. The best overall solution is kept as the result. A *GRASPxELS* can be seen as a metaheuristic that captures good features of pure greedy algorithms (intensification), of random construction procedures and of mutation (diversification). In order to avoid premature convergence of iterative search process efficient clone detection is included trying to prevent unprofitable exploration of search space previously investigated. To measure difference between two schedules earliest starting time of operations and CO_2 emissions at each schedule are compared.

Algorithm 1: GRASPxELS metaheuristic principles

```
Procedure name GRASP×ELS
Begin
    S* ← Ø
    for p := 1 to np do
        S ← Construction_Phase
        S ← Local_Seach_Phase_Makespan
        S ← Local_Seach_Phase_CO2
        if( f(S) < f(S*)) then S* ← S end if
        for j:= 1 to ni do                        // ELS phase
            f″:=+∞            // initialized best found solution
            for k := 1 to ns do                   // ELS mutation
              S′:= S
              Mutate S′
              S′ ← Local_Seach_Phase_Makespan
              S′ ← Local_Seach_Phase_CO2
              if (f(S′) < f″) then
                  f″ := f(S′); S″ := S′
              end if
            end for
            if (f″ < f(S*)) then S* ← S″ end if
            S := S″           // update ELS current solution
        end for
    end for
    return S*
end
```

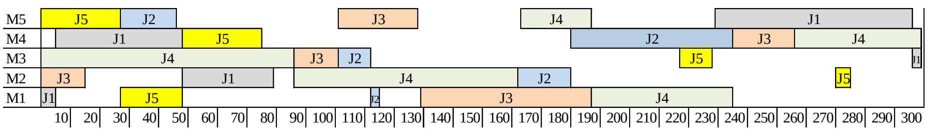

Fig. 1. Gantt chart of schedule when makespan is minimized (C_{max}=299, CO_2=95776)

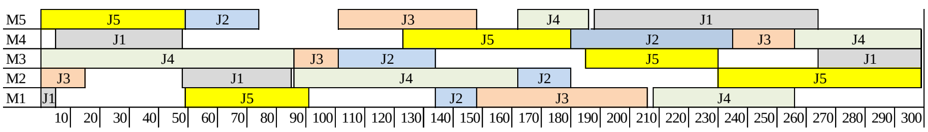

Fig. 2. Gantt chart of schedule when CO_2 emissions are minimized (C_{max}=299, CO_2=95512)

4 Computational experiment

4.1 Instances

The benchmark is concerned with instances based on the OR-library which instances encompass the well-known Laurence's instances La01-LA10. From these 10 instances of Laurence considered, we have generated 60 instances of *JSSM*. From each original LA instance we built six instances with 5, 6 …10 jobs. Each operation has three modes of treatment, and each operation mode has its own duration. The duration of the first mode is exactly the same of the duration of the classical Job Shop. The durations of the two others modes are randomly generated around the value of the first mode. To generate the CO_2 generated by operations in our model, we considered that the generation of CO_2 is inversely proportional to the processing time of each operation. That's can be explained by the fact that if we have to make a choice between three ways of dealing with different durations, our choice will be the mode of treatment with the shortest processing time, but with this mode treatment, a higher consumption of carbon is observed, due to the high speed imposed on the production line. For example, if the set is composed by three operations with duration 10, 40 and 50 units times, the carbon consumed by these operations are respectively proportional at the values (10+40+50)/10 = 10, (10+40+50)/40 = 5 and (10+40+50)/50 = 2.

4.2 Computational results

The GRASP×ELS is benchmarked over the 60 generated instances and it is compared with solutions obtained by linear programming. All procedures are implemented under Delphi 6.0 package and experiments were carried out on a 2.8 GHz computer under Windows 7 with 12 GO of memory. The results for 13 instances whose optimal solutions are obtained using CPLEX are given in Table 1. As we make ten replications, the average value of the makespan, CO_2 and CPU times are considered in Table 1. For these instances our approach found ten optimal solutions concerning the makespan objective and an average deviation from the optimal solutions of 0.17%. Whit an average deviation of 0.08% from the CO_2 objective optimal solutions we can state that the quality achieved is very good. Solutions are found in rather short computational time (in average of 1.20 s), which proves that GRASPxELS is a powerful method. To estimate the quality of proposed framework we compared the results with the lower bounds (*LB*) calculated by the linear programming with CPLEX 24 hours' time limit. For 33 instances lower bounds are obtained and the average improvement of the lower bounds is 29.53% for makespan and 37.33% for the CO_2. For the rest of the instances (14) no CPLEX solutions are obtained during time limit. Moreover, all 60 solutions (job sequences) obtained by our *GRASPxELS* are compared with those obtained by linear programming as follows: for each best job sequence generated by *GRASPxELS* linear program found the optimal solution concerning the makespan and the CO_2. CPLEX results are compared with these obtained by our metaheuristic. An analysis of comparative results showed that the average deviations are 0.00% and 0.025% respectively from the makespan and CO_2. Therefore, the performance evaluation of the proposed framework clearly shows that it is a particularly efficient method for problem under study.

Table 1. Results of comparative study: optimal solutions

Linear programming optimal solutions						*GRASPxELS* solutions				
Inst.	*n*	*m*	C_{max}	CO_2	*T* (secs)	*BFS*	C_{max} *Dev%*	*Avg.* CO_2	T(s)	CO_2 *Dev%*
La01	5	5	300	98258	873.25	300.0	0.00	98419.2	0.09	0.16
La01	6	5	333	136929	1504.49	333.0	0.00	136877.1	0.22	0.04
La01	8	5	395	288853	10509.00	401.7	1.67	288970.0	6.16	0.04
La02	5	5	299	95562	1246.08	299.0	0.00	95637.4	0.09	0.08
La02	6	5	342	138536	1121.71	342.9	0.26	138732.0	2.08	0.14
La02	7	5	307	192610	5685.50	307.0	0.00	192702.9	2.46	0.05
La03	5	5	290	96489	1548.49	290.0	0.00	96513.5	0.09	0.03
La03	6	5	283	134679	20460.00	283.0	0.00	134831.1	0.17	0.11
La03	7	5	299	192135	14889.00	301.0	0.76	192163.5	3.41	0.01
La04	5	5	291	82133	1219.41	291.0	0.00	82264.5	0.06	0.16
La04	6	5	322	116491	1487.37	322.0	0.00	116580.0	0.41	0.08
La05	6	5	286	129235	5840.88	286.0	0.00	129369.3	0.17	0.10
La05	7	5	398	179891	19429.00	398.0	0.00	180196.0	0.22	0.17
						Average	0.17%		1.20	0.08%

5 Conclusion

Many real life problems can be modelled as a job-shop scheduling problem where machines can work at different speeds. It represents an extension of the classical job-shop scheduling problem, where each operation has to be executed by one machine and this machine has the possibility to work at different speeds. In this paper, we analyse the relationship among two important objectives that must be taken into account in green manufacturing: Makespan and Energy-efficiency. To solve the problem we propose an efficient GRASPxELS algorithm. The algorithm was evaluated on 60 test problems based on well-known La01, …, La10 instances and was shown to produce optimal or near-optimal solutions on all instances. The numerical experiment proves that our framework can obtain almost optimal solutions in a rather short computational time. In this paper the makespan is one the objective to minimize, however the structure of the proposed GRASPxELS permits to minimize other criteria like: maximum tardiness or total flowtime, … This work is a step forward definition of efficient models for job-shops like scheduling problems with multiple speed machines.

References

1. Salido, M.A., Escamilla, J., Barber, F., Giret, A., Tang, D., Dai, M.: Energy-aware Parameters in Job-shop Scheduling Problems. In GREEN-COPLAS 2013: IJCAI 2013 Workshop on Constraint Reasoning, Planning and Scheduling Problems for a Sustainable Future, pp. 44-53, Beijing, China (2013)

2. Garey, M. R., Johnson, D. S., Seth, R.: The complexity of flowshop and jobshop scheduling. Math. of Oper. Res. 1, 117-129 (1976)
3. Jain, S., Meeran, S.: Deterministic job-shop scheduling: Past, present and future. . E. J. of Oper. Res. 113, 390-434 (1999)
4. Gao, L., Zhang., D., Zhang, L, Li, X.: An efficient memetic algorithm for solving the job shop scheduling problem. C. & Ind. Eng. 60, 699-705 (2011)
5. Chassaing, M., Fontanel, J., Lacomme, P., Ren, L., Tchernev, N., Viullechenon, P.: A GRASP × ELS approach for the job-shop with a web service paradigm packaging. Exp. S. with Appl., 41(2), 544-562 (2014)
6. He, Y., Lui, F., Cao, H.J., Li, C.: A bi-objective model for job-shop scheduling problem to minimize both energy consumption and makespan. J. Cent. South Uni. Techn. 12(2), 167-171 (2005)
7. Liu, Y., Dong, H., Lohse, N., Petrovic, S., Gindy, N.: An investigation into minimizing total energy consumption and total weighted tardiness in job shops. J. Clean. Prod., 65, 87-96 (2014)
8. Yildirim, M.B., Mouzon, G., Single-machine sustainable production planning to minimize total energy consumption and total completion time using a multiple objective genetic algorithm. IEEE Tr. on Eng. Man. 59(4), 585-597 (2011)
9. Fang, K., Uhan, N., Zhao, F. Sutherland, J. Flow shop scheduling with peak power consumption constraints. A. of Op. Res. 206(1), 115-145 (2013).
10. Luo, H., Huang, G., Chen, H., Li, X. Hybrid flow shop scheduling considering machine electricity consumption cost. I. J. Prod. Ec. 146, 423–439 (2013)
11. Bruzzone, A.A.G., Anghinolfi, D, Paolucci, M., Tonelli, F. Energy-aware scheduling for improving manufacturing process sustainability: a mathematical model for flexible flowshops. CIRP Ann.—Man. Techn. 61(1), 459–62 (2012)
12. Prins C.: A GRASP x evolutionary local search hybrid for the vehicle routing problem. In: Pereira, F.B., Tavares, J. (eds).: Bio-inspired algorithms for the vehicle routing problem, Studies in computational intelligence vol. 61, pp.35-53, Springer-Verlag, Berlin Heidelberg (2009).
13. Feo, T.A., Resende, M.G.C.: A probabilistic heuristic for a computationally difficult set covering problem. Op. Res. Let. 8, 67–71 (1989)
14. Feo, T.A., Resende, M.G.C.: Greedy randomized adaptive search procedures. J. of Gl. Opt. 6, 109–133, (2005)
15. Feo, T.A., Resende, M.G.C., Smith, S.H.: A greedy randomized adaptive search procedure for maximum independent set. Op. Res. 42, 860–878 (1994)
16. Resende, M.G.C., Ribeiro, C.: Greedy randomized adaptive search procedures. In: Glover F., Kochenberger G. (eds). ORMS, Handbook of metaheuristics, pp. 219-249, Kluwer Academic Publishers, Dordrecht (2003)
17. Festa, P., Resende, M.G.C.: An annotated bibliography of GRAS part I: algorithms. Int. Trans. in Op. Res. 16, 1–24 (2009)
18. Festa, P., Resende, M.G.C.: An annotated bibliography of GRASP part II: applications. Int. Trans. in Op. Res. 16, 131–172 (2009)
19. Roy B., Sussmann B., Les problèmes d'ordonnancement avec contraintes disjunctive, In : Note DS N°9 bis, SEMA, Paris, 1964.
20. Bierwirth, C.: A Generalized permutation approach to Job Shop scheduling with genetic algorithms. OR Spektrum, 17, 87-92 (1995)
21. Van Laarhoven, P.J.M., Aarts, E.H.L., Lenstra, J.K.: Jobshop scheduling by simulated annealing, Op. Res. 40(1):113-125 (1992)