A Real-Time Optimization Algorithm for the Integrated Planning and Scheduling Problem Towards the Context of Industry 4.0

Mário Leite
University of Minho
Centro ALGORITMI
School of Engineering
Department of Production and Systems Engineering
Portugal

Telmo Pinto
University of Minho
Centro ALGORITMI
School of Engineering
Department of Production and Systems Engineering
Portugal

Cláudio Alves
University of Minho
Centro ALGORITMI
School of Engineering
Department of Production and Systems Engineering
Portugal

In this paper, we analyze the integration of two different problems in the supply chain, concerning the tactical and operational levels, and how the integration of two complex problems can be profitable towards the context of industry 4.0. More precisely, we address the integrated planning and scheduling problem on parallel and identical machines, seeking fast solutions that are globally optimal and flexible.
In the planning phase, a set of jobs is assigned to their processing periods of time. On the other hand, in the scheduling phase, jobs are assigned to a machine in a given order.
We propose a new metaheuristic approach through a variable neighborhood descent algorithm which iteratively explores four neighborhood structures with a first improvement strategy.
The suggested algorithm was extensively tested using a large set of benchmark instances. The obtained results are discussed and compared with other approaches from literature.

Keywords: *industry 4.0, supply chain, planning, scheduling, optimization algorithms, metaheuristics.*

1. INTRODUCTION

The integrated optimization of supply chain processes is a meaningful way to achieve better solutions to the problems that are increasingly demanding and complex. The economic and financial uncertainties of markets may cause variations in the production chain, which requires fast and effective reactions not only to the isolated processes but also throughout the chain using dynamic and highly flexible plants [1].

Nowadays, besides seeking the optimization of processes inside the companies, there is also growing concern about the outer consequences of each decision, from a more holistic standpoint [2-4]. In this context, gathering sustainability with production is well-recognized [5,6].

Sustainability is seen as an essential part of supply chains. Sustainable thinking and acting through a green supply chain, environmentally friendly decision-making, while keeping or reducing costs, may allow the industry and companies to achieve competitive business advantages [7]. Traditional supply chains become insufficient to meet all requirements, and thus new strategies are mandatory [8].

Currently, the transition to the fourth industrial revolution is taking place through the potentialities of Cyber-Physical Systems (CPS) and the Internet of Things (IoT) that make factories smarter, more sustainable and more inclusive [9,10].

The "Industry 4.0" allowed a close connection between the physical and the virtual world, leading to more flexibility to variations, reduced lead times, more customization ability and cost reduction.

The existence of multiple and highly sensitive devices able to collect and send information to other devices connected to the network can allow almost instantaneous intelligent actions and enable them to make other analysis [11, 12]. Typical examples include devices incorporated in processing machines, in products, or even in pieces as parts of the finished product.

In [13, 14], the authors also emphasize the ability to exchange information not only between internal processes but also with suppliers and customers (external to the organization), possibly leading to more reliable and more representative data. Recognition of the needs for each phase of the supply chain in real time, often with no human interference [15], opens new windows of opportunities.

With a significant increase of information sources, data processing in large scale and control mechanisms are also necessary, since unprocessed information cannot help in decision-making and it only consumes resources and time [16]. The information should be oriented to support decisions or alert to problems and issues which were unknown or imperceptible until then.

In [17], the authors mention that the existence of a combination of big data tools and decision support systems would be a perfect match. However, despite the improvement of the inputs that support the optimization algorithms, there is still a significant lack of technological tools to explore the full potential of "Industry 4.0", and therefore, more sophisticated methods are required [18].

Received: November 2018, Accepted: May 2019
Correspondence to: Dr. Telmo Pires Pinto
Department of Production and Systems Engineering.
University of Minho, Guimarães. Portugal
E-mail: telmo@dps.uminho.pt
doi:10.5937/fmet1904775L

In this sense, developing methods that simultaneously integrate different processes of the supply chain is one of the tendencies pointed out in [19] as a way to obtain more flexible solutions. The development of these methods as tools to better support decision-making in smaller companies may be essential to balance the fact that these companies may be behind in the digitalization process (due to the costs and time that the entire process can take [17]).

In this sense, we propose new approaches to a complex problem that simultaneously considers two of the most significant processes in supply chain production: planning and scheduling. In the literature, and not surprisingly, these problems tend to be analyzed separately since they both are NP-hard. However, this separation of strongly related processes may lead to a failure when attempting to obtain globally optimal solutions [20, 21]. To find good solutions in a reduced computational time, we suggest heuristic and metaheuristic algorithms based on local search and neighborhood changes through a variant of variable neighborhood search [22]. Our aim is to provide a real-time optimization algorithm in order to obtain global and acceptable solutions in real-time to face unpredicted events that can require re-optimization.

We resort to the definition of the integrated planning and scheduling problem (IPSP) in parallel and identical machines described in [23]. The problem has a defined time horizon, divided into equal periods of time, in which jobs have to be processed, and each one has a release date and a due date. The planning phase defines which jobs are processed in each period of time, while the scheduling phase assigns and orders the jobs in the available machines. For each job, there are penalty factors for early or late processing time every time a job is not processed at its due date. The overall objective of the IPSP is to find the planning and scheduling solution that minimizes the sum of all penalties. In order to assess the quality of suggested approaches, all the results were compared to other contributions relying on exact methods [24] and on a simple local search heuristic [25].

This paper is organized as follows. In Section 2, we define the integrated planning and scheduling problem. In Section 3, the metaheuristic based approach is described in detail. The description of the benchmark instances, the obtained results and a critical analysis are presented in Section 4. The main conclusions are drawn in Section 5.

2. INTEGRATED PLANNING AND SCHEDULING PROBLEM

The IPSP can be formulated as follows. The entire time horizon T is divided into τ smaller periods of time ($T = \{1,\ldots,\tau\}$) of equal size $P \in \mathbb{N}$. There is a set of jobs N that should be processed in a set of parallel and identical machines M. Each job $j \in N$ has five attributes:

- Processing time ($p_j \in$ □), corresponding to the units of time required to process job j on any machine;
- Release date ($r_j \in T$), corresponding to the first period of time since when job j is available to be processed;
- Due date ($d_j \in T$) is the period of time when job j must be processed without penalties;
- Penalty factor for earliness ($e_j \in$ □): penalty to be paid for each period that job j is processed in advance to its due date: if job j is processed in a period $t \in T$ such that $t < d_j$, the resulting penalty is equal to $e_j\,(d_j - t)$;
- Penalty factor for lateness ($l_j \in$ □): penalty to be paid for each period that job j is processed after its due date: if job j is processed in a period $t \in T$ such that $t > d_j$, the resulting penalty is equal to $l_j\,(t - d_j)$.

Therefore, the weight of the penalty (w_j^t) of a job $j \in N$ processed in a period of time $t \in T$, is given as follows:

$$w_j^t = e_j \times \max\left\{0, d_j - t\right\} + l_j \times \max\left\{0, t - d_j\right\}. \quad (1)$$

Additionally, each machine can only process one job at a time, and each job can only be processed on one machine in a single period of time. Thus, it is not possible to interrupt a job processing, even between consecutive periods of time. As a consequence, the time required to process each job should be less than or equal to the maximum size of the period ($p_j \leq P$, □$j \in N$).

The objective of IPSP is to determine the period to process each job while minimizing the sum of penalties.

3. A VARIABLE NEIGHBORHOOD DESCENT BASED APPROACH

In this section, we present the metaheuristic based approach aiming to achieve good solutions in a relatively short computational time. Firstly, the constructive heuristic to obtain the initial solutions is described. Then, neighboring structures applied in the local search phase are enumerated and described. Finally, we describe all the elements of the variable neighborhood descent (VND) algorithm.

3.1 Initial Solutions

In order to obtain starting solutions, we resort on one of the constructive heuristics proposed in [26] which consists of iteratively assigning each job of the set N to the set of available machines M using the following steps:

1. Sort the set of jobs in decreasing order of the ratio of the sum of the penalty factors by the processing time:

$$(e_j + l_j)/p_j . \quad (2)$$

In case of a tie, priority is given to jobs with longest processing time.

2. Assign each job to its due date if possible. That is, a job $j \in N$ is assigned to its due date if there is any available machine $m \in M$ with the capacity to process it ($p_j \leq capacity_t^m$, where $capacity_t^m \in$ □ corresponds to the free capacity of period $t \in T$ of machine $m \in M$).
3. The remaining non-allocated jobs preserve its ordering. Each job will be assigned to a previous or a following period with a free capacity to process it while minimizing the penalty to be paid for antici-

pation or delay. In this case, the value of the objective function will always be positive since it is not possible to assign at least one job in its due date.

3.2. Neighborhood Structures

In order to obtain neighborhood solutions in the metaheuristic approach, four different neighborhood structures were defined, as follows.

N_1 – Inserting one job: the neighbors of a given solution consist of all obtained solutions by selecting one job assigned to a given period of time and moving it to another period.

N_2 - Swapping two jobs from different periods: the neighbors of a given solution consist of all obtained solutions by exchanging two jobs assigned to different periods of time.

N_3 - Swapping two different lists of jobs from different periods: the neighbors of a given solution consist of all obtained solutions by selecting two ordered lists of jobs not assigned to the same period, and by exchanging these two lists while the order of jobs in each list remains unchanged. The lists of jobs can be assigned to the same machine or not.

N_4 - Swapping two jobs from the same period in different machines, and inserting another job in the same period. The neighbors of a given solution consist of all obtained solutions by firstly exchanging two jobs at a given period, and then inserting a given third job in the same period. In the first step, the selected pair of jobs must be assigned to the same period at different machines while the third job must be assigned to a different period (i.e., other than the considered one at the first step).

3.3 Variable Neighborhood Descent Algorithm

An exclusively deterministic variant of the basic variable neighborhood search (VNS) was developed. More precisely, the variable neighborhood descent method (VND) is proposed to tackle the IPSP. This metaheuristic explores several different neighborhood structures in the local search process so that it can achieve a global optimal solution more efficiently than a simple local search with a single neighborhood structure. A first improvement strategy is adopted in the local search phase. The four neighborhood approaches referred to above ($N = \{N_1, N_2, N_3, N_4\}$) are sequentially explored (basic sequential variable neighborhood descent procedure [27]). That is, if there is an improvement, a new iteration is performed using the new improved solution, and the local search is applied through the first neighborhood structure. Otherwise, the following neighborhood structure is explored, or a local optimum was reached.

For the sake of clarity, the VND algorithm is summarized in Figure 1. In this algorithm, the input is given by a benchmark instance of the IPSP (*Instance*), a maximum number of neighborhood structures (k_{max}) and a set N. A feasible solution of the IPSP is represented by S and *z(S)* corresponds to its value of the objective function. The first solution S is obtained through the constructive heuristic described in subsection 3.1, and denoted by *ConstructiveHeuristic(Instance)*. The local search phase is denoted by *SimpleLocalSearch(S, N_k)* and it explores the neighborhood structure N_k, with $1 \leq k \leq k_{max}$.

Algorithm 1. Variable Neighborhood Descent Algorithm

```
Function VND (Instance, k_max, N)
S := ConstructiveHeuristic (Instance);
k := 1;
while k ≤ k_max and z(S) > 0 do
    S' := SimpleLocalSearch(S, N_k);
    if z(S') < z(S) then
        S := S';
        k := 1;
    end
    else
        k := k+1;
    end
end
return S
```

4. COMPUTATIONAL EXPERIMENTS

In order to evaluate the VND algorithm, we resort on an extensive set of benchmark instances for the IPSP [23]. The computational experiments were performed on a PC with Intel Core i7, 2.70GHz processor and 8GB of RAM.

4.1 Benchmark Instances

The set of benchmark instances is divided into three large sets according to the processing time of jobs, as follows:

- set A which is composed of only the so-called small jobs (jobs with processing time between 1 and 33 units of time);
- set B with half small jobs and the other half with large jobs (jobs with processing time between 34 and 100 units of time);
- set C includes only instances with large jobs.

The capacity of each period is equal to 100 units of time (P = 100) in all instances. For all sets, there are 5 different instances for each combination of the number of jobs ($|N|$), the number of available machines ($|M|$), and the value τ_0 that defines the maximum possible variation between release date and due date of jobs (which may be between period 1 and period τ_0). In order to be possible to solve all instances in the time horizon, the value of the number of available periods (τ) is defined in [23] by:

$$\tau = \tau_0 + \left\lceil 2 \cdot \frac{\sum_{j \in N} p_j}{P \cdot |M|} \right\rceil . \quad (3)$$

Therefore, a feasible solution is always ensured.

The number of machines and the value of τ_0 are the same for all sets: $|M| \in \{2,6,10\}$ and $\tau_0 \in \{2,6,10\}$. The number of jobs is different for different sets. For the set A, $|N| \in \{100, 150, 200, 250, 300\}$ while for the set B,

$|N| \in \{50, 100, 150, 200, 250, 300\}$. Finally, for the set C, $|N| \in \{40, 60, 80, 100\}$. Therefore, considering 5 instances for each possible combination results in a total of 675 test instances (225 instances for set A, 270 in set B, and 180 for set C).

4.2 Obtained Results and Discussion

Based on the obtained results, we perform a critical analysis comparing them with other approaches from literature [24, 25]. Note that the exact methods proposed in [24] have a computational time limit of 1200 seconds. Additionally, lower and upper bounds were provided, which are used in this analysis. Note also that in [25], a simple local search heuristic with first improvement strategy is presented, where the same set of neighorhood structures *N* was explored, but in a separate way.

In Tables 1 and 2, we report on the total (*tot.*) or the average (*avg.*) obtained results for all sets. The overall results are grouped by number of jobs and machines in Table 1, and by number of jobs and τ_0 in Table 2, as in [23, 24]. In both tables, each row represents the average or total values for each group of 15 instances.

Table 1. Computational results by the number of jobs and number of machines

			Exact Methods					Simple LS				VND			
set	$\|N\|$	$\|M\|$	*bestLB*	*bestUB*	*opt*	*gap*	*t*	*bestLS*	*opt*	*gap*	*t*	*bestVND*	*opt*	*gap*	*t*
A	100	2	446.13	768.93	0	0.31	1200.68	566.93	0	0.32	0.00	505.73	0	0.22	0.02
	100	6	35.73	35.93	14	0.00	112.09	44.87	7	0.21	0.00	41.20	7	0.18	0.00
	100	10	7.00	7.00	15	0.00	7.63	13.60	11	0.13	0.00	8.60	11	0.05	0.00
A	150	2	1432.00	3356.60	0	0.61	1205.04	1763.27	0	0.23	0.00	1572.53	0	0.13	0.04
	150	6	127.80	160.33	8	0.05	614.31	172.67	3	0.33	0.00	149.33	3	0.22	0.00
	150	10	38.07	38.13	15	0.00	142.68	54.80	9	0.15	0.00	43.47	9	0.08	0.00
A	200	2	2975.73	12102.73	0	0.75	1203.98	3634.27	0	0.22	0.01	3218.60	0	0.11	0.06
	200	6	345.00	714.07	5	0.30	855.55	451.80	0	0.38	0.00	388.33	0	0.23	0.01
	200	10	96.33	133.20	11	0.05	390.69	127.53	10	0.08	0.00	112.47	10	0.05	0.00
A	250	2	5176.40	23173.73	0	0.76	1203.87	6115.93	0	0.17	0.02	5506.40	0	0.07	0.12
	250	6	678.47	1893.33	0	0.63	1201.01	877.00	0	0.33	0.00	767.53	0	0.22	0.03
	250	10	207.00	416.53	10	0.16	528.57	276.87	5	0.27	0.00	230.93	5	0.15	0.01
A	300	2	8496.87	63203.00	0	0.85	938.67	9840.20	0	0.14	0.04	8880.73	0	0.05	0.26
	300	6	1185.00	3481.80	0	0.72	1205.33	1562.67	0	0.37	0.01	1319.67	0	0.21	0.08
	300	10	415.47	1192.93	5	0.45	865.55	544.87	3	0.31	0.00	471.73	3	0.16	0.03
set A	tot./avg.		1444.20	7378.55	83	0.38	778.38	1736.48	48	0.24	0.01	1547.82	48	0.14	0.04
B	50	2	337.73	346.07	15	0.04	75.36	522.73	0	0.41	0.00	426.60	0	0.27	0.00
	50	6	30.60	31.33	15	0.02	2.00	49.33	5	0.30	0.00	38.40	5	0.23	0.00
	50	10	8.73	8.93	15	0.01	0.91	16.53	10	0.16	0.00	11.87	10	0.08	0.00
B	100	2	1949.00	1980.13	2	0.02	1134.81	2765.40	0	0.31	0.00	2297.07	0	0.17	0.02
	100	6	282.00	285.40	12	0.03	276.44	437.13	0	0.48	0.00	362.27	0	0.33	0.01
	100	10	88.80	89.60	15	0.02	15.41	137.53	4	0.35	0.00	114.93	4	0.26	0.00
B	150	2	5497.67	7122.60	0	0.16	1201.37	7381.40	0	0.26	0.01	6246.73	0	0.13	0.07
	150	6	870.53	892.73	3	0.04	1075.50	1354.60	0	0.44	0.00	1103.93	0	0.28	0.03
	150	10	327.87	330.13	12	0.03	358.60	505.47	0	0.48	0.00	404.93	0	0.31	0.01
B	200	2	10808.67	23009.53	0	0.53	1202.47	13824.67	0	0.22	0.03	11923.13	0	0.10	0.22
	200	6	1993.67	2269.93	0	0.14	1200.72	2869.67	0	0.36	0.01	2342.40	0	0.19	0.09
	200	10	708.33	733.47	4	0.06	948.24	1080.93	0	0.49	0.01	873.27	0	0.31	0.04
B	250	2	18099.20	38926.13	0	0.53	853.94	22686.00	0	0.21	0.07	19681.53	0	0.08	0.46
	250	6	3569.00	4415.67	0	0.23	1160.77	4911.13	0	0.31	0.03	4132.00	0	0.16	0.21
	250	10	1398.73	1503.40	0	0.10	1200.74	2060.00	0	0.41	0.01	1681.67	0	0.24	0.11
B	300	2	26644.20	53841.47	0	0.23	57.65	34189.53	0	0.22	0.11	29157.87	0	0.09	0.82
	300	6	5760.13	20966.40	0	0.53	238.56	7889.20	0	0.29	0.05	6621.20	0	0.15	0.38
	300	10	2444.33	3046.20	0	0.19	390.16	3528.73	0	0.37	0.03	2855.67	0	0.20	0.25
set B	tot./avg.		4489.96	8877.73	93	0.16	632.98	5900.56	19	0.34	0.02	5015.30	19	0.20	0.15
C	40	2	741.13	741.13	15	0.00	0.11	841.47	0	0.14	0.00	770.73	1	0.06	0.00
	40	6	86.07	86.07	15	0.00	0.05	97.60	2	0.18	0.00	91.60	5	0.13	0.00
	40	10	20.73	20.73	15	0.00	0.03	25.20	7	0.15	0.00	22.27	9	0.08	0.00
C	60	2	1868.40	1868.40	15	0.00	0.36	2060.13	0	0.11	0.00	1905.60	1	0.03	0.01
	60	6	291.40	291.40	15	0.00	0.10	338.47	0	0.24	0.00	305.80	1	0.10	0.00
	60	10	91.67	91.67	15	0.00	0.07	109.60	3	0.23	0.00	97.07	4	0.13	0.00
C	80	2	3779.27	3779.27	15	0.00	1.25	4080.67	0	0.08	0.00	3834.53	0	0.02	0.02
	80	6	653.27	653.27	15	0.00	0.25	764.13	0	0.21	0.00	683.13	1	0.09	0.01
	80	10	218.80	218.80	15	0.00	0.15	264.33	0	0.30	0.00	232.60	2	0.15	0.00
C	100	2	6473.73	6473.73	15	0.00	3.06	6944.00	0	0.07	0.01	6533.80	0	0.01	0.05
	100	6	1140.80	1140.80	15	0.00	0.39	1303.13	0	0.18	0.01	1183.07	0	0.07	0.02
	100	10	420.80	420.80	15	0.00	0.24	497.20	0	0.28	0.00	443.93	0	0.12	0.02
set C	tot./avg.		1315.51	1315.51	180	0.00	0.50	1443.83	12	0.18	0.00	1342.01	24	0.08	0.01

Table 2. Computational results by the number of jobs and $\tau 0$

			Exact Methods					Simple LS				VND			
set	$\|N\|$	τ_0	*bestLB*	*bestUB*	*opt*	*gap*	*t*	*bestLS*	*opt*	*gap*	*t*	*bestVND*	*opt*	*gap*	*t*
A	100	2	379.93	472.20	9	0.06	512.06	452.93	1	0.24	0.00	411.07	1	0.11	0.01
	100	6	89.13	316.07	10	0.21	403.82	135.40	7	0.26	0.00	112.60	7	0.21	0.00
	100	10	19.80	23.60	10	0.05	404.52	37.07	10	0.16	0.00	31.87	10	0.13	0.00
A	150	2	1059.27	1702.27	5	0.18	933.91	1262.93	0	0.22	0.00	1128.53	0	0.10	0.04
	150	6	391.73	996.67	8	0.21	609.43	512.47	4	0.27	0.00	452.13	4	0.16	0.01
	150	10	146.87	856.13	10	0.28	418.68	215.33	8	0.22	0.00	184.67	8	0.17	0.00
A	200	2	2136.93	4773.80	1	0.37	1174.63	2512.20	0	0.19	0.01	2253.67	0	0.09	0.04
	200	6	908.07	4231.20	5	0.41	809.17	1171.53	5	0.19	0.00	1007.07	5	0.11	0.02
	200	10	372.07	3945.00	10	0.32	466.41	529.87	5	0.29	0.00	458.67	5	0.19	0.01
A	250	2	3406.73	9660.47	0	0.51	1204.95	3972.20	0	0.18	0.01	3577.47	0	0.07	0.07
	250	6	1738.87	7558.60	5	0.54	917.28	2107.33	0	0.36	0.01	1876.20	0	0.21	0.05
	250	10	916.27	8264.53	5	0.49	811.23	1190.27	5	0.22	0.00	1051.20	5	0.15	0.03
A	300	2	5336.27	21769.40	0	0.54	1160.13	6178.27	0	0.16	0.02	5527.53	0	0.06	0.16
	300	6	2956.00	20369.00	0	0.86	1160.56	3536.47	0	0.33	0.02	3155.60	0	0.19	0.12
	300	10	1805.07	25739.33	5	0.63	688.87	2233.00	3	0.32	0.01	1989.00	3	0.18	0.08
set A	tot./avg.		1444.20	7378.55	83	0.38	778.38	1736.48	48	0.24	0.01	1547.82	48	0.14	0.04
B	50	2	253.27	256.40	15	0.02	34.91	360.53	0	0.37	0.00	290.73	0	0.18	0.00
	50	6	85.20	88.73	15	0.02	36.54	147.00	5	0.32	0.00	122.80	5	0.27	0.00
	50	10	38.60	41.20	15	0.02	6.82	81.07	10	0.18	0.00	63.33	10	0.13	0.00
B	100	2	1261.33	1266.53	11	0.01	439.37	1633.07	0	0.26	0.00	1417.20	0	0.15	0.01
	100	6	662.13	676.73	8	0.04	574.73	1027.27	0	0.49	0.00	837.07	0	0.33	0.01
	100	10	396.33	411.87	10	0.03	412.56	679.73	4	0.40	0.00	520.00	4	0.28	0.01
B	150	2	3299.13	3615.40	4	0.03	950.19	4139.27	0	0.24	0.01	3678.47	0	0.13	0.04
	150	6	1935.13	2830.93	3	0.11	1003.96	2860.93	0	0.43	0.00	2326.93	0	0.25	0.04
	150	10	1461.80	1899.13	8	0.09	681.32	2241.27	0	0.51	0.00	1750.20	0	0.33	0.03
B	200	2	6108.47	9841.93	1	0.18	1132.94	7523.73	0	0.21	0.02	6607.87	0	0.09	0.14
	200	6	4313.00	8714.47	0	0.25	1201.18	5860.73	0	0.37	0.02	4927.07	0	0.21	0.12
	200	10	3089.20	7456.53	3	0.29	1017.31	4390.80	0	0.49	0.01	3603.87	0	0.30	0.09
B	250	2	10064.20	16612.67	0	0.20	1086.96	12225.60	0	0.19	0.04	10795.13	0	0.08	0.31
	250	6	7423.27	15551.80	0	0.29	1111.94	9583.67	0	0.31	0.03	8260.53	0	0.17	0.25
	250	10	5579.47	12680.73	0	0.37	1016.56	7847.87	0	0.42	0.03	6439.53	0	0.24	0.21
B	300	2	14913.33	42534.67	0	0.30	263.10	18429.87	0	0.20	0.07	16056.67	0	0.08	0.58
	300	6	11110.00	16591.40	0	0.26	143.09	14831.93	0	0.30	0.06	12371.07	0	0.14	0.48
	300	10	8825.33	18728.00	0	0.39	280.19	12345.67	0	0.39	0.05	10207.00	0	0.22	0.39
set B	tot./avg.		4489.96	8877.73	93	0.16	632.98	5900.56	19	0.34	0.02	5015.30	19	0.20	0.15
C	40	2	494.47	494.47	15	0.00	0.07	547.67	0	0.11	0.00	499.73	3	0.03	0.00
	40	6	241.07	241.07	15	0.00	0.06	267.07	3	0.20	0.00	256.93	5	0.14	0.00
	40	10	112.40	112.40	15	0.00	0.06	149.53	6	0.16	0.00	127.93	7	0.10	0.00
C	60	2	1210.53	1210.53	15	0.00	0.22	1294.47	0	0.09	0.00	1222.73	2	0.02	0.01
	60	6	665.53	665.53	15	0.00	0.12	755.93	0	0.23	0.00	691.60	1	0.09	0.00
	60	10	375.40	375.40	15	0.00	0.19	457.80	3	0.26	0.00	394.13	3	0.15	0.00
C	80	2	2258.80	2258.80	15	0.00	0.57	2395.93	0	0.08	0.00	2272.93	1	0.01	0.02
	80	6	1484.93	1484.93	15	0.00	0.59	1654.27	0	0.23	0.00	1527.87	0	0.09	0.01
	80	10	907.60	907.60	15	0.00	0.48	1058.93	0	0.28	0.00	949.47	2	0.16	0.01
C	100	2	3698.13	3698.13	15	0.00	1.57	3921.67	0	0.07	0.01	3715.20	0	0.01	0.04
	100	6	2583.47	2583.47	15	0.00	1.01	2858.47	0	0.19	0.01	2630.87	0	0.07	0.03
	100	10	1753.73	1753.73	15	0.00	1.11	1964.20	0	0.28	0.01	1814.73	0	0.13	0.02
set C	tot./avg.		1315.51	1315.51	180	0.00	0.50	1443.83	12	0.18	0.00	1342.01	24	0.08	0.01

The approaches referred to above are denominated by "Exact Methods" and "Simple LS" ([24] and [25], respectively), while VND algorithm is denominated by "VND". The meaning of the columns of the table is defined as follows:

- *set*: set of instances with common features related to the processing time of jobs (set A, B or C);
- $|N|$: number of jobs;
- $|M|$: number of machines;
- τ_0: nominal length of the time horizon;
- *bestLB*: the average of best lower bound given in [24];
- *bestUB*: the average of best upper bound given in [24];
- *bestLS*: the average of the best value obtained by local search heuristics in [25];
- *bestVND*: the average of the best solutions obtained through the VND algorithm;
- *opt*: number of instances solved up to optimality;
- *gap*: the average of the optimality gap between the best result for each approach and the lower bound (column *bestLB*). For instance, let vLB_i be the lower bound value and $vVND_i$ the best result of VND approach for a given instance i. The *gap* value is given by $(vVND_i - vLB_i) / vVND_i$;
- *t*: average computational time to achieve the final solution (in seconds).

The presented results for set A show that VND approach presents average values that are closer to the lower bound (*bestLB*) with an average gap of 14%. The

VND algorithm was able to outperform both approaches, local search and exact method, with average optimality gaps of 24% and 38%, respectively. However, the number of instances solved up to optimality does not follow this trend. The VND approach was able to achieve only 48 optimal solutions against 83 through exact methods. Nevertheless, the exact approach tends to achieve best average results for instances with a lower number of jobs and a high number of machines (100 or 150 jobs and 6 or 10 machines), as it is depicted in Figure 2.

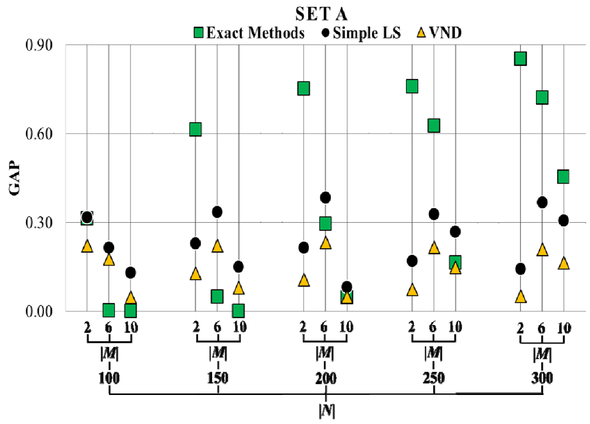

Figure 2. Average of optimality gap in set A

Concerning set B, the VND algorithm was able to achieve the best solutions among all approaches in terms of the average value of the objective function, and thus the average upper bound was reduced. However, VND was only able to outperform the average of optimality gap of the local search approach (20% against 34%) whereas the exact method reduced this value to 16%. Again, the VND approach was not able to outperform the number of optimal solutions.

In set C, it was also possible for the VND approach to improve the local search solutions: the number of optimal solutions was doubled, and there was a decrease of 10% of the average gap. Not surprisingly, VND was not able to outperform the exact method in this set since all instances were solved up to optimality using an exact algorithm.

It is worth noting that the required computational time to obtain a solution is significantly different between VND and the exact methods, being one of the most important advantages of the implemented methods in this paper. While the exact methods take an average of hundreds of seconds to find the final solution, VND spent only hundredths of a second.

5. CONCLUSIONS

In this paper, we explored the integrated planning and scheduling problem on parallel and identical machines. This problem integrates two NP-hard problems belonging to different levels of supply chain management.

We resort on heuristics and metaheuristics to tackle this problem. More precisely, we defined a variable neighborhood descent algorithm as well as four neighborhood structures, seeking to obtain fast solutions. All the elements of the algorithm were implemented, and an exhaustive set of computational experiments was conducted using benchmark instances.

The proposed algorithm was able to outperform a local search heuristic proposed in the literature. Additionally, it was able to be competitive with exact methods for instances composed of a large number of jobs and a lower number of machines, within only hundredths of a second towards a real-time optimization algorithm in the context of industry 4.0.

ACKNOWLEDGMENTS

This work has been supported by FCT – Fundação para a Ciência e a Tecnologia within the Project Scope: UID/CEC/00319/2019.

REFERENCES

[1] Pfeffer, J., Graube, M., Reipschlaeger, P., Arndt, S., Urbas, L., Dachselt, R., & Stelzer, R. (2015). Towards collaborative plant control using a distributed information and interaction space. In *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)* (pp. 1-4). IEEE.

[2] Yüksek, L., Kaleli, H., Özener, O., & Özoğuz, B. (2009). The Effect and Comparison of Biodiesel-Diesel Fuel on Crankcase Oil, Diesel Engine Performance and Emissions. FME Transactions. 37, 91-97.

[3] Rajković, M., Zrnić, N., Kosanić, N., Borovinšek, M., & Lerher, T. (2017). A Multi-Objective Optimization model for minimizing cost, travel time and CO2 emission in an AS/RS. FME Transactions, 45(4), 620–629.

[4] Tuček, D., Tučková, Z., & Jelínková, D. (2017). Performance Measurement of Energy Processes in Czech Production Plants. FME Transactions, 45(4), 670–677.

[5] Edtmayr, T., Sunk, A., & Sihn, W. (2016). An approach to integrate parameters and indicators of sustainability management into value stream mapping. *Procedia CIRP*, 41, 289-294.

[6] Kleindorfer, P. R., Singhal, K., & Van Wassenhove, L. N. (2005). Sustainable operations management. *Production and operations management*, 14(4), 482-492.

[7] Gerlitz, L. (2016). Design management as a domain of smart and sustainable enterprise: business modelling for innovation and smart growth in Industry 4.0. *Entrepreneurship and Sustainability Issues*, *3*(3), 244-268.

[8] Chen, B., Wan, J., Shu, L., Li, P., Mukherjee, M., & Yin, B. (2018). Smart factory of industry 4.0: key technologies, application case, and challenges. IEEE Access, 6, 6505-6519.

[9] Shafiq, S. I., Sanin, C., Szczerbicki, E., & Toro, C. (2015). Virtual engineering object/virtual engineering process: a specialized form of cyber physical system for Industrie 4.0. *Procedia Computer Science*, *60*, 1146-1155.

[10] Ungurean, I., Gaitan, N. C., & Gaitan, V. G. (2014, May). An IoT architecture for things from industrial environment. In *2014 10th International Conference on Communications (COMM)* (pp. 1-4). IEEE.

[11] Wan, J., Cai, H., & Zhou, K. (2015). Industrie 4.0: Enabling technologies. In *Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things* (pp 135–140). IEEE.

[12] Trappey, A., Trappey, C., Fan, C., Hsu, A., Li, X., & Lee, I. (2017). IoT patent roadmap for smart logistic service provision in the context of Industry 4.0. *Journal of the Chinese Institute of Engineers*, *40*(7), 593-602.

[13] Abdel-Basset, M., Manogaran, G., & Mohamed, M. (2018). Internet of Things (IoT) and its impact on supply chain: A framework for building smart, secure and efficient systems. *Future Generation Computer Systems*, *86*, 614-628.

[14] Grangel-González, I., Baptista, P., Halilaj, L., Lohmann, S., Vidal, M. E., Mader, C., & Auer, S. (2017, September). The industry 4.0 standards landscape from a semantic integration perspective. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (pp. 1-8). IEEE.

[15] Ivanov, D., Dolgui, A., Sokolov, B., Werner, F., & Ivanova, M. (2016). A dynamic model and an algorithm for short-term supply chain scheduling in the smart factory industry 4.0. *International Journal of Production Research*, *54*(2), 386-402.

[16] Cheng, Y., Chen, K., Sun, H., Zhang, Y., & Tao, F. (2018). Data and knowledge mining with big data towards smart production. *Journal of Industrial Information Integration*, *9*, 1-13.

[17] Trstenjak, M., & Cosic, P. (2017). Process planning in Industry 4.0 environment. *Procedia Manufacturing*, *11*, 1744-1750.

[18] Xu, L., Xu, E., & Li, L. (2018). Industry 4.0: state of the art and future trends. *International Journal of Production Research*, *56*(8), 2941-2962.

[19] Stank, T., Autry, C., Daugherty, P., & Closs, D. (2015). Reimagining the 10 megatrends that will revolutionize supply chain logistics. *Transportation Journal*, *54*(1), 7-32.

[20] Chang, Y. C., & Lee, C. Y. (2004). Machine scheduling with job delivery coordination. *European Journal of Operational Research*, *158*(2), 470-487.

[21] Chu, Y., & You, F. (2014, December). Integrated planning and scheduling by hybrid solution method. In *53rd IEEE Conference on Decision and Control* (pp. 6818-6823). IEEE.

[22] Hansen, P., & Mladenović, N. (2003). Variable neighborhood search. *Handbook of Metaheuristics*, 24(11), 145–184.

[23] Kis, T., & Kovács, A. (2012). A cutting plane approach for integrated planning and scheduling. *Computers and Operations Research*, 39(2), 320–327.

[24] Rietz, J., Alves, C., Braga, N., & Valério de Carvalho, J. (2016). An exact approach based on a new pseudo-polynomial network flow model for integrated planning and scheduling. *Computers and Operations Research*, 76, 1–22.

[25] Leite, M., Pinto, T., Alves, C. (2017). A local search heuristic for integrated process planning and scheduling. *Eighth International Conference on Business Sustainability 2018*, Póvoa do Varzim, Portugal (submitted).

[26] Leite M., Alves C., Pinto T. (2017) Variable Neighborhood Search for Integrated Planning and Scheduling. Lecture Notes in Computer Science, vol 10405. Springer, Cham.

[27] Hansen, P., Mladenović, N., Todosijević, R., & Hanafi, S. (2017). Variable neighborhood search: basics and variants. *EURO Journal on Computational Optimization*, *5*(3), 423-454.

АЛГОРИТАМ ОПТИМИЗАЦИЈЕ У РЕАЛНОМ ВРЕМЕНУ ЗА ПРОБЛЕМ ИНТЕГРИСАНОГ ПЛАНИРАЊА И ПРОГРАМИРАЊА ПРОИЗ-ВОДЊЕ У КОНТЕКСТУ ИНДУСТРИЈЕ 4.0

М. Леите, Т. Пинто, К. Алвеш

У овом раду анализирана је интеграција два различита проблема у ланцу снабдевања, у вези са тактичким и оперативним нивоима, и како интеграција два комплексна проблема може бити профитабилна у контексту Индустрије 4.0. Прецизније, рад се бави интегрисаним проблемом планирања и програмирања производње на паралелним и идентичним машинама, тражећи брза решења која су глобално оптимална и флексибилна.
У фази планирања, одређени број радних мјеста је додељен њиховим временима процесирања. С друге стране, у фази планирања, задаци се додељују машини у датом редоследу. Предлажен је нови метахеуристички приступ кроз алгоритам опадајућих варијабилних непосредних структура који итеративно истражује четири непосредно блиске структуре стратегијом првог побољшања. Предложени алгоритам је екстензивно тестиран коришћењем великог броја „бенчмарк“ примерака. Добијени резултати су продискутовани и упоређивани са другим приступима из литературе.