Heuristic approaches for lot splitting and scheduling in identical parallel machines

Carina Pimentel (carina@dps.uminho.pt)
Centro de Investigação Algoritmi, Universidade do Minho, Braga, Portugal

Filipe Pereira e Alvelos
DPS/Centro de Investigação Algoritmi, Universidade do Minho, Braga, Portugal

José M. Valério de Carvalho
DPS/Centro de Investigação Algoritmi, Universidade do Minho, Braga, Portugal

António Duarte
Departamento de Gestão Industrial, Instituto Politécnico de Bragança, Bragança, Portugal/Centro de Investigação Algoritmi, Universidade do Minho, Braga, Portugal

Abstract

In this paper we address a practical lot splitting and scheduling problem of a textile company that produces fine knitted garments. The problem consists of finding a weekly production plan for the knitting section, in which the garment components are produced in a set of parallel machines. We solve the problem in two steps using heuristic approaches. In the first step one of two heuristics (a network flow heuristic and a constructive heuristic) is applied to find an initial solution and in the second step local search based algorithms are applied to improve the quality of the solutions.

Keywords: Production Lot Splitting and Scheduling, Heuristics, Real-World problem

Introduction

In this study we propose methods to deal with a production planning problem of a textile industry that produces fine knitted goods, such as cardigans, pants, dresses, sweaters and scarves. The problem is related with the knitting production process in which the main components (garment parts) of a product are made. The aim is to develop a tool to simultaneously solve two production planning problems of the knitting section: i) a lot splitting problem in which the components demand is divided into several smaller size lots, to speed up production and ii) an assignment and scheduling problem, in which the lots determined in i) are assigned and scheduled in a set of parallel machines.

A weekly production plan has to be prepared, establishing the production lots for each component, where (what machines) and when to produce them. As usual in this kind of problem, there are multiple, possibly conflicting, objectives. For this particular problem there are two important objectives: on time delivery of garments and minimum levels of work-in-process inventory. To evaluate a production plan we use a function that weights the following two measures: (1) total tardiness of products and (2) total deviation occurred during the production of each product. The total deviation of a product is the sum of all the deviations of each component lot completion time from the product completion time. The product completion time is the completion time of the component lot that finishes last. Although in the context of the real problem, the first objective has higher priority, the second one is very important to assure a smooth processing flow. The next production stage, after the components knitting, is joining the several components that belong to the same product. Since this process can only occur after completing all the components production, their completing times should ideally be the same. As tardiness has greater negative impacts, its weight is much higher than the deviation weight.

The parallel machine lot splitting and scheduling problem with time based objectives did not yet been adequately studied in the literature. We are not aware of any work considering our second objective. In recent review papers related to scheduling (see for example (Allahverdi et al., 2008) and (Zhu and Wilhelm, 2006)) the authors mention that research addressing due date related objectives need to be emphasized.

Our problem can not be untied from the classical parallel machine scheduling problem (PMSP), in which there are *n* jobs to schedule in *m* machines aiming at optimizing one or more performance measures. However, there are two important differences: in our problem a given job (component) can be split into several lots of smaller size and can be processed in more than one machine at the same time, while in PMSP a job can not be partitioned or preempted; also, in our problem a job (product) is divided into several sub-jobs (components) that are related to each other because the job completion time depends of the completion times of all the sub-jobs, while in PMSP jobs are independent of each other. Two review papers addressing enumerative and approximate algorithms for PMSP are the ones of Cheng and Sin (1990) and Mokotoff (2001).

The identical parallel machine scheduling problem with job splitting, without setup times and with objective to minimize total tardiness is NP-hard (Xing and Zhang, 2000). As our problem is an extension of the previous one, it is also NP-hard. Tahar et al. (2006) study the identical parallel machine scheduling problem with job splitting, sequence dependent setups and with objective to minimize makespan. The authors use a two step heuristic algorithm to solve the problem. There are several differences between our problem and the above one: (1) the objectives are different; (2) their problem has sequence dependent setups while our problem does not have setups; (3) in their problem jobs are independent of each other, while our problem has sub-jobs (components) that are associated with jobs (products); and (4) in their problem every job can be done in every machine, while our problem restricts job assignments to specific machines. Sheen and Liao (2007) consider a preemptive scheduling problem, for identical parallel machines, with availability constraints, and with objective to minimize maximum lateness. They solve the problem using a series of maximum flow problems. The main differences between this problem and our are: (1) in our problem a job can be split, meaning that it can be divided into several smaller lots that may be produced at the same time in different machines, while in their problem, a job can be preempted, but not be processed at the same time in different machines; (2) the objectives are different, and (3) their jobs are independent of each other.

We solve the lot splitting and scheduling problem (LSSP) by approximate methods. We use a network flow heuristic and a constructive heuristic, which were previously proposed in Pimentel et al. (2010a) and Pimentel et al. (2010b), to quickly find solutions to the problem. We propose four local search algorithms and two metaheuristics based on systematic changes of the neighborhoods used by the local search algorithms: a variable neighborhood search (Hansen and Mladenovic, 2001) and a variable neighborhood descent (Hertz and Mittaz, 2001). The methods are compared using a set of generated instances, similar to real-world ones.

The main contributions of this work are: i) the treatment of a lot splitting and scheduling problem that, besides splitting demands into lots of smaller size and sequencing those lots in a set of parallel machines, determines the beginning and finishing instant times of each lot in each machine, thus allowing the modeling of the new objective related to the minimization of the time deviation between the completion time of a product and the completion time of all the components lots belonging to it; ii) the development and comparison of a set of methods, some very fast and some more time consuming but providing better solutions, that provide solutions for this NP-hard problem allowing automated scheduling for a knitting section of a textile industry.

Problem definition

We begin this section by giving a brief description of the company. It produces about 1.300.000 finished knitted goods per year, distributed among an average of 4.300 different products. The factory purchases yarns and transform them into garment pieces through four productive sections: knitting, linking, dyeing and finishing. The problem we are dealing with belongs to the knitting section, in which the yarns are transformed into component pieces, mostly sleeves, back bodies, front bodies and scarves. Each product is composed by a set of those components that will be joined in the linking section.

The production system of the knitting section is organized in three groups of parallel machines. Each group has a gauge associated and there is a unique relationship between the gauge of a machine and the product. Because of that, the scheduling plans must be prepared by gauge. Within a gauge the machines are identical, since they take the same amount of time to produce one unit of a given component. In addition, for each gauge, there is a 0/1 compatibility matrix between the machines and the components, that specifies the machines where a given component can be produced. This association matrix is needed due to technical reasons.

The production plans for the section are prepared every week and in each plan several production orders are considered. Each production order has information about the products (defined by a piece of cloth and size) requested by customers, as well as their due dates, quantities requested and the corresponding set of components. For each component a unit production time is known. The completion time of a product is the completion time of the last component produced. Since all the work associated with the production orders is available for processing at the beginning of the planning horizon, the components do not have release dates. However, each machine has a given release date. The setup times involved in a change between garment parts are neglected.

In order to accelerate the production process, the quantities requested of each component can be split into several lots, and each of those lots is assigned to a given machine. Two or more lots of the same component may be assigned to the same machine (possibly, scheduling lots of other components between them) and idle intervals may exist between two consecutive lots. Besides that, several lots of the same component can be processed at the same time in different machines.

Currently, the lot splitting decisions are taken by the planning department, prior to be sent to the knitting manager. The lots are created by product and range between 800 to 999 pieces. Each of those lots is further divided by the number of product sizes requested, proportionally to the quantities requested in each size. The assignment and scheduling decisions are taking manually by the knitting manager, based on common sense rules and on the several years of experience.

Initial solution

In this section we briefly describe two heuristics used to find an initial solution for the proposed local search algorithms and metaheuristics. The first approach consists of solving two network flow problems combined with a scheduling procedure. The second approach is a list scheduling constructive heuristic that explores specific characteristics of LSSP.

Network flow heuristic

The network flow heuristic for the LSSP has two steps. In both steps a network flow model is solved and the flow solution obtained is transformed into a valid schedule using a simple single pass procedure. The scheduling procedure is the same in both steps, but the network models have some differences. The origins of the networks correspond to components to be produced, and the destinations correspond to time intervals in a given machine. A solution to the network flow problem states how much to produce of each component in each time interval/machine. The reader is referred to Pimentel et al. (2010a) for additional details about the network flow heuristic.

List scheduling heuristic

A list scheduling algorithm is a constructive heuristic that determines a schedule for a given ordering of jobs. Our list scheduling algorithm runs three steps to define an initial solution. In the first step, an ordered list of products is defined. In the second step, an ordered list of components is defined, taking into account the ordered list of step one. In step three the components of the ordered list of step two are iteratively selected one by one, and assigned and scheduled in one or more machines. In Pimentel et al. (2010b) the list scheduling algorithm is presented in detail.

Local search based algorithms

Basic local search algorithms

A local search (LS) algorithm, is an improvement algorithm, that starts from some initial solution and iteratively tries to replace the current solution by a better one, in an appropriated defined neighborhood of the current solution (see, for example, (Blum and Roli, 2003), for an introduction to LS). In Anderson et al. (1997) LS is deeply explored in machine scheduling context.

The two alternative methods described in the previous section can be used to start the local search process. We choose the method with better global performance in a set of test instances, presented in the next section. A wise selection of the neighborhood structure is very important, as it defines the modifications that are allowed in the current solution. We define four neighborhood structures that take into account specific characteristics of the particular problem at hands. All of them are based on lot insertions. In an insertion move a lot is removed from its current position and is inserted at another position (a position is defined by a machine and a completion time instant). The position completion time instant is defined by an objective date. The rationale of the concept of objective dates lies in the aim of minimizing product deviations. The objective dates set of a given product are given by its due date, the starting times of all its components lots in the current solution, and the completion times of the same components lots. In each neighborhood structure the search is conducted over all lots, all machines and all objective dates, in a first descent strategy. In each move a given lot is inserted in a given machine finishing at a given objective date.

The four neighborhood structures are: total back insert (TBI), partial back insert (PBI), total ahead insert (TAI) and partial ahead insert (PAI).

We illustrate the four neighborhood structures using a small example. Consider a problem with two machines and two products, A and B. Product A, has due date equal to 32, and has two components: A1 with duration 10 and A2 with duration 13. Product B, has due date equal to 32 and has two components too: B1 with duration 6 and B2 with duration 9. Further consider the current solution depicted in Figure 1. In Figure 2 a neighbor solution of the current one is presented for each of the four neighborhood structures.

The TBI solution represented in Figure 2(a), corresponds to a move from the current solution selecting lot A2 of M1 to be inserted in M2, finishing at objective date 26 (completion time of lot A1 in M1). In TBI, when the insertion forces one or more lots of the current solution to be rescheduled, those lots are totally moved backward. The change of lot A2 of M1, forced lots A2 and B2 of M2 to be totally moved backward. The partial back insertion move presented in Figure 2(b), corresponds to an insertion of lot A2 of M1 in M2 with objective date 26. In PBI only the parts of the lots of the current solution that are occupied by the lot to be inserted, are moved backward. That is why in Figure 2(b), only three units of lot B2 are moved backward. Note that this change forces lot B2 to be split into two lots. TAI is similar to TBI and PAI similar to PBI, but in TAI and PAI the lots are moved ahead instead of backward. The solution of TAI of Figure 2(c) corresponds to an insertion of lot A2 of M1 in M2, finishing at objective date 26. This move forces lot B2 of M2 to be totally moved ahead and lot A2 is batched. The PAI solution of Figure 2(d) corresponds to the insertion of lot A2 of M1 in M2 with objective date equal to 32 (due date of product A). Note that during the local search process the size of the neighborhoods can change, because the number of lots may increase (due to splits of lots) or decrease (due to batches of lots). Also, the number of objective dates is dynamic too.

The evaluation function of the four LS procedures is the one previously presented, *i.e.*, to minimize total tardiness of products and to minimize total deviations occurred during production. The typical representation of a scheduling solution in LS algorithms is as a permutation of jobs. Based on this permutation, the complete solution (including starting and completion times) can be easily obtained. In our problem, this solution representation is not adequate, because lots can be split, two or more lots of the same component may be assigned to the same machine and idle intervals may exist between two consecutive lots. Our solutions are represented through a set of lots, storing for each lot its starting time, duration time, component associated and machine associated.

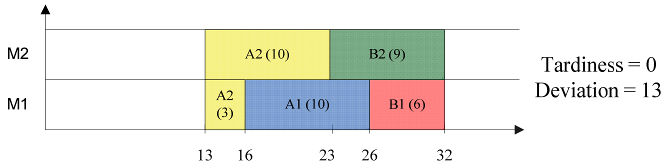
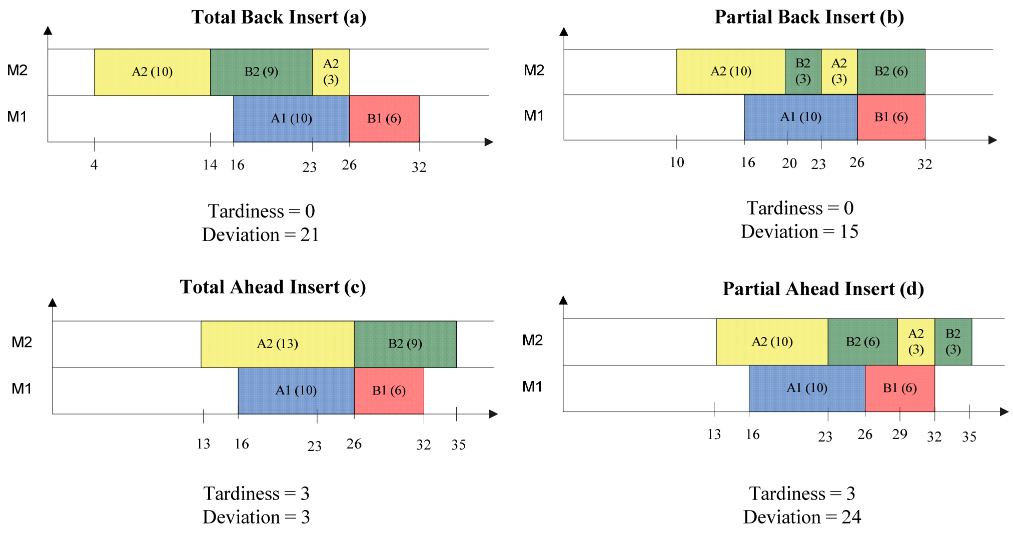


Figure 1 – Current solution of example



Total Back Insert (a)

Tardiness = 0
Deviation = 21

Partial Back Insert (b)

Tardiness = 0
Deviation = 15

Total Ahead Insert (c)

Tardiness = 3
Deviation = 3

Partial Ahead Insert (d)

Tardiness = 3
Deviation = 24

Figure 2 – Neighbors of current solution of example

Metaheuristics

A well known drawback of LS algorithms is its inability to escape from local optimum solutions. During the last decades, new heuristic algorithms, that incorporates mechanisms to prevent the algorithm to stop at local optimum solutions, have emerged. Those heuristic algorithms are usually termed by metaheuristics. In Blum and Roli (2003) the most popular metaheuristics are fully examined. The book edited by Xhafa and Abraham (2008) presents several metaheuristic approaches for scheduling problems arising in industrial and manufacturing applications. VNS and VND are two metaheuristics that use systematic changes of neighborhood to avoid local optimum entrapment. When in a given neighborhood structure the search reaches a local minimum, the algorithm switches to a different neighborhood structure, re-starting the search process. VND is a deterministic algorithm in which the current solution is used as starting point to a new search process after a change of neighborhood. On the other hand, VNS is a stochastic process. In this case, when there is a change of a neighborhood structure the algorithm starts from a randomly generated solution, which belongs to the neighborhood of the current solution.

In our study we apply a sequential VNS and a sequential VND, that combine the four neighborhood structures presented in the previous sub-section. Taking into account the local search results, that will be presented in the next section, the neighborhood search order (for VNS and VND) chosen was PBI-TBI-PAI-TAI. The stopping condition for VND is the current solution be a local optimum for the four neighborhood structures. For VNS, the algorithm stops if after searching two times all the neighborhood structures a better solution is not found. The neighborhood structures used within VNS to randomly generate the solutions to start the search are the same used during the search process. For additional details of VNS or of VND see (Hansen and Mladenovic, 2001) and (Blum and Roli, 2003).

Computational experiments

In this section we test all the algorithms presented in the previous sections using a set of 54 instances, randomly generated taking into account data obtained at the company. The set of 54 instances is divided by gauge into three groups (18 instances of gauge 21, 18 of gauge 27 and 18 of gauge 24). All the algorithms were coded in visual C++, and the tests were run in a personal computer. We set a time limit of 2 hours in all the algorithms. The network flow models are solved with Cplex 11.0 (ILOG, 2007). In Table 1 we present the instances sizes (measured in terms of number of components and number of machines), and some performance measures for the initial solutions, obtained with the network flow heuristic and with the list scheduling heuristic. In the network flow heuristic several machine orders are tested, and the algorithm selects the best one. In column 7 of Table 1 we present the time needed to solve the problem with the best machine order while in column 8 the time to test all the possible (including the best one) machine orders. The machine utilization of a given machine *m* is given by:

$$\frac{\textit{Total occupied time of } m}{\textit{Time Horizon - Release date of } m} \times 100 \; .$$

The list scheduling heuristic is clearly superior to the network flow heuristic, both in solution times and in solution quality. Note that orders due dates were generated randomly, as were the quantities requested, meaning that in some of the test instances, it may not be possible to finish on time a given order, even if all the resources were assigned to it.

Table 1 – Instances size, and initial solutions results

					Solution time (seconds)			Total tardiness (hours)		Total deviation (hours)		Number of lots		Average
Instance	Gauge	Number of products	Number of components	Number of machines	List scheduling heuristic	Network flow heuristic best order	Network flow heuristic all orders	List scheduling heuristic	Network flow heuristic	List scheduling heuristic	Network flow heuristic	List scheduling heuristic	Network Flow Heuristic	machine utilization (%)
Inst20T1.1.G21	21	8	18	5	0	0	0.219	5.60	5.60	254.36	6.28	23	19	46.3
Inst20T1.2.G21	21	9	25	5	0	0.031	0.031	3.38	64.88	1094.73	2260.82	41	47	102.1
Inst20T1.3.G21	21	9	17	5	0	0.047	0.093	20.98	236.79	622.91	1089.67	45	30	80.7
Inst30T1.1.G21	21	18	50	5	0	0.047	0.047	77.52	299.17	2208.37	2841.54	117	91	103.9
Inst30T1.2.G21	21	20	56	5	0	0.249	0.249	2.26	3.24	533.33	726.26	78	91	68.6
Inst30T1.3.G21	21	20	60	5	0	0.047	0.047	41.42	148.52	755.74	796.76	132	101	92.7
Inst40T1.1.G21	21	20	58	5	0	0.031	0.031	0.00	0.00	609.20	2779.85	77	95	99.4
Inst40T1.2.G21	21	24	69	5	0	0.047	0.047	5.50	51.49	1109.36	2781.88	112	121	101.9
Inst40T1.3.G21	21	27	74	5	0	0.015	0.047	0.00	72.40	267.97	1272.74	101	107	79.3
Inst50T1.1.G21	21	29	74	5	0	0.031	0.031	0.00	48.03	395.91	3049.83	94	116	87.6
Inst50T1.2.G21	21	30	74	5	0	0.031	0.078	7.56	39.87	1483.99	1800.21	114	127	104.5
Inst50T1.3.G21	21	33	99	5	0	0.046	0.078	0.00	0.22	629.01	1917.34	129	168	86.4
Inst60T1.1.G21	21	26	70	5	0	0.047	0.047	0.00	0.00	1207.00	2989.24	99	113	98.0
Inst60T1.2.G21	21	30	77	5	0	0.047	0.047	38.92	68.28	977.02	954.94	114	116	66.4
Inst60T1.3.G21	21	29	77	5	0	0.063	0.109	15.82	31.84	475.85	3004.03	121	131	96.0
Inst70T1.1.G21	21	30	90	5	0	0.046	0.171	114.34	411.61	578.99	1698.32	250	130	59.3
Inst70T1.2.G21	21	33	83	5	0	0.047	0.078	0.00	0.00	933.28	1963.57	106	133	60.1
Inst70T1.3.G21	21	39	116	5	0	0.047	0.047	0.00	231.76	345.41	3191.78	140	169	79.2
Inst20T1.1.G27	27	31	89	11	0.015	0.172	2.262	0.00	0.00	836.37	4609.06	132	253	88.1
Inst20T1.2.G27	27	32	84	11	0	0.126	2.683	15.84	124.71	1247.35	1820.89	178	244	89.1
Inst20T1.3.G27	27	29	84	11	0	0.108	15.085	2.20	6.73	1211.50	3764.61	120	221	63.9
Inst30T1.1.G27	27	41	107	11	0	0.233	22.698	0.00	0.00	454.94	4667.33	159	304	82.0
Inst30T1.2.G27	27	38	103	11	0	0.203	53.695	40.67	268.05	2461.20	5804.63	372	263	105.9
Inst30T1.3.G27	27	44	128	11	0.015	0.202	28.065	0.00	0.72	1061.65	6841.38	183	348	95.6
Inst40T1.1.G27	27	53	142	11	0.016	0.702	74.708	0.00	0.00	938.34	8168.44	191	370	95.7
Inst40T1.2.G27	27	42	112	11	0	0.358	40.529	0.00	0.29	1077.09	5570.20	174	291	95.4
Inst40T1.3.G27	27	47	125	11	0.015	0.202	4.634	34.28	94.44	1964.49	1654.74	289	301	75.9
Inst50T1.1.G27	27	43	120	11	0	0.14	13.322	0.00	0.00	1668.75	5594.34	196	260	99.5
Inst50T1.2.G27	27	65	174	11	0.016	0.577	2191.96	16.80	80.43	1595.13	6403.38	368	480	103.6
Inst50T1.3.G27	27	60	154	11	0.016	0.438	36.223	0.00	1.13	646.32	8804.44	204	451	83.7
Inst60T1.1.G27	27	71	197	11	0	0.436	817.534	0.00	0.00	1857.14	4389.20	264	481	99.7
Inst60T1.2.G27	27	76	210	11	0.031	0.983	26.52	298.15	1199.88	1727.42	12866.60	901	575	80.2
Inst60T1.3.G27	27	67	181	11	0.016	0.483	1947.85	94.50	306.93	1535.01	7602.79	486	488	96.9
Inst70T1.1.G27	27	70	182	11	0.016	0.343	9.672	361.68	222.31	2239.87	5370.20	700	393	97.5
Inst70T1.2.G27	27	81	221	11	0.031	1.372	166.639	251.45	593.43	2339.09	5582.84	728	551	98.5
Inst70T1.3.G27	27	67	187	11	0.016	0.702	34.226	146.32	578.99	2362.96	7834.71	651	511	91.3
Inst20T1.1.G24	24	34	94	13	0	0.202	228.185	0.00	0.90	500.37	5501.33	148	269	95.6
Inst20T1.2.G24	24	37	108	13	0.015	0.187	183.112	22.29	74.17	2256.61	8195.83	287	358	91.8
Inst20T1.3.G24	24	34	98	13	0	0.188	568.604	126.32	549.57	2205.99	4848.91	426	298	96.6
Inst30T1.1.G24	24	51	139	13	0	0.5	7013.06	0.00	0.00	816.46	6228.37	195	370	82.7
Inst30T1.2.G24	24	49	135	13	0	0.406	778.97	2.89	19.41	2832.25	10131.40	240	441	101.8
Inst30T1.3.G24	24	38	90	13	0	0.187	11.778	30.54	137.33	2082.22	4314.71	271	313	97.5
Inst40T1.1.G24	24	57	152	13	0	0.593	196.139	1.88	26.39	2364.74	5248.89	235	457	91.2
Inst40T1.2.G24	24	64	174	13	0	0.405	117.234	0.00	0.37	499.41	5231.96	226	474	82.5
Inst40T1.3.G24	24	57	157	13	0.016	0.748	998.899	98.31	184.42	2008.03	5265.22	569	440	94.8
Inst50T1.1.G24	24	55	152	13	0.015	0.562	60.091	0.00	1.50	322.24	3396.84	204	424	65.4
Inst50T1.2.G24	24	70	197	13	0.031	0.624	1174.04	426.55	1260.94	2606.13	16225.40	906	501	114.0
Inst50T1.3.G24	24	70	184	13	0.016	0.546	448.609	6.05	6.92	1403.93	6791.82	378	584	96.3
Inst60T1.1.G24	24	81	216	13	0.016	0.936	1515.88	0.00	0.00	797.37	10081.70	284	603	90.5
Inst60T1.2.G24	24	69	188	13	0.015	0.842	1503.54	12.10	78.97	3239.33	15931.60	327	633	103.9
Inst60T1.3.G24	24	81	232	13	0.015	1.856	4255.31	0.96	14.28	2476.19	15523.80	335	685	100.6
Inst70T1.1.G24	24	82	226	13	0.016	0.765	265.684	0.00	1.00	973.29	13740.50	295	661	88.5
Inst70T1.2.G24	24	94	254	13	0.016	1.981	2825.5	99.09	118.67	2720.66	5362.57	746	716	109.8
Inst70T1.3.G24	24	108	277	13	0.031	2.776	3582.59	17.04	70.18	2342.35	13840.60	752	903	101.5

We use the same set of instances to compare the four LS algorithms with each other, and to compare them with the initial solution (obtained with the list scheduling heuristic). The algorithms start the search from the list scheduling heuristic solution. The results are presented in Table 2. We skip over the total tardiness results from Table 2, as the LS algorithms are not able to reduce total tardiness. Only in instance Inst70T1.3.G24 PAI reduces total tardiness in 2.5 hours. In 47 of the 54 instances PBI has longer solution times, compared to the other LS algorithms, but it always improves total deviation. The average improvement by instance is 43.3%. TBI, PAI and TAI, improve total deviation in 48, 48 and 31 instances, respectively. Moreover, the average improvement by instance is 20.4%, 9.4% and 6%, respectively. We mark for each instance, with red color, the algorithm that further improves total deviation. In general, all the LS algorithms reduce the number of lots of a solution, when compared with the initial solution.

Table 2 – List scheduling heuristic and basic local search algorithms results

	Solution time (seconds)				Total deviation (hours)					Average number of lots per component					Average deviation per product				
Instance	PBI	TBI	PAI	TAI	List scheduling heuristic	PBI	TBI	PAI	TAI	List scheduling heuristic	PBI	TBI	PAI	TAI	List scheduling heuristic	PBI	TBI	PAI	TAI
Inst20T1.1.G21	0.27	0.14	0.05	0.05	254.36	7.79	11.07	50.94	53.17	1.28	1.11	1.11	1.11	1.11	31.80	0.97	1.38	6.37	6.65
Inst20T1.2.G21	0.14	0.08	0.05	0.05	1094.73	695.97	1089.03	1094.73	1094.73	1.64	1.60	1.60	1.64	1.64	121.64	77.33	121.00	121.64	121.64
Inst20T1.3.G21	0.09	0.05	0.06	0.05	622.91	456.27	622.91	592.19	622.91	2.65	2.59	2.65	2.59	2.65	69.21	50.70	69.21	65.80	69.21
Inst30T1.1.G21	3.89	0.25	0.42	0.20	2208.37	1091.35	2192.96	2148.55	2208.37	2.34	2.34	2.32	2.32	2.34	122.69	60.63	121.83	119.36	122.69
Inst30T1.2.G21	1.42	0.98	0.56	0.45	533.33	328.25	421.44	406.54	481.89	1.39	1.32	1.27	1.30	1.34	26.67	16.41	21.07	20.33	24.09
Inst30T1.3.G21	2.65	0.45	0.27	0.27	755.74	412.24	753.97	755.44	755.44	2.20	2.13	2.20	2.20	2.20	37.79	20.61	37.70	37.77	37.77
Inst40T1.1.G21	1.75	1.28	0.25	0.19	609.20	360.48	448.68	563.67	606.73	1.33	1.28	1.29	1.31	1.33	30.46	18.02	22.43	28.18	30.34
Inst40T1.2.G21	3.96	0.45	2.17	0.30	1109.36	850.16	1106.06	1061.33	1109.36	1.62	1.65	1.61	1.61	1.62	46.22	35.42	46.09	44.22	46.22
Inst40T1.3.G21	4.91	5.48	0.94	0.45	267.97	157.29	159.78	231.89	237.51	1.36	1.26	1.27	1.31	1.32	9.92	5.83	5.92	8.59	8.80
Inst50T1.1.G21	20.26	10.97	0.73	0.39	395.91	173.63	208.84	327.16	371.56	1.27	1.19	1.16	1.28	1.27	13.65	5.99	7.20	11.28	12.81
Inst50T1.2.G21	8.60	0.62	0.56	0.33	1483.99	592.59	1476.72	1440.32	1483.99	1.54	1.43	1.53	1.53	1.54	49.47	19.75	49.22	48.01	49.47
Inst50T1.3.G21	28.25	105.38	4.45	3.99	629.01	276.07	275.64	516.06	516.07	1.30	1.17	1.14	1.24	1.24	19.06	8.37	8.35	15.64	15.64
Inst60T1.1.G21	20.44	11.01	2.37	1.70	1207.00	760.65	887.27	1138.00	1138.29	1.41	1.36	1.29	1.39	1.37	46.42	29.26	34.13	43.77	43.78
Inst60T1.2.G21	3.31	0.53	0.64	0.64	977.02	631.99	976.00	976.80	843.43	1.48	1.43	1.48	1.48	1.45	32.57	21.07	32.53	32.56	28.11
Inst60T1.3.G21	16.26	12.14	0.94	0.27	475.85	207.63	292.01	475.52	475.85	1.57	1.48	1.44	1.56	1.57	16.41	7.16	10.07	16.40	16.41
Inst70T1.1.G21	9.61	3.42	3.34	0.80	578.99	423.69	576.43	564.39	578.78	2.78	2.74	2.74	2.74	2.77	19.30	14.12	19.21	18.81	19.29
Inst70T1.2.G21	33.87	11.56	2.84	1.87	933.28	113.14	395.10	343.91	483.68	1.28	1.16	1.19	1.23	1.23	28.28	3.43	11.97	10.42	14.66
Inst70T1.3.G21	24.96	23.29	3.03	2.17	345.41	207.72	160.52	255.35	339.60	1.21	1.19	1.14	1.18	1.20	8.86	5.33	4.12	6.55	8.71
Inst20T1.1.G27	70.78	75.91	4.37	4.45	836.37	262.17	299.33	584.97	591.87	1.48	1.35	1.31	1.44	1.44	26.98	8.46	9.66	18.87	19.09
Inst20T1.2.G27	43.15	4.10	29.70	1.61	1247.35	519.89	1195.73	1098.89	1247.35	2.12	2.12	2.10	2.12	2.12	38.98	16.25	37.37	34.34	38.98
Inst20T1.3.G27	11.54	2.17	1.50	1.47	1211.50	397.28	1211.47	1211.48	1211.48	1.43	1.43	1.43	1.43	1.43	41.78	13.70	41.77	41.78	41.78
Inst30T1.1.G27	60.23	66.47	8.33	8.27	454.94	284.93	215.23	412.43	412.43	1.49	1.49	1.35	1.45	1.45	11.10	6.95	5.25	10.06	10.06
Inst30T1.2.G27	103.88	5.55	5.74	5.65	2461.20	1584.96	2461.20	2461.20	2461.20	3.61	3.57	3.61	3.61	3.61	64.77	41.71	64.77	64.77	64.77
Inst30T1.3.G27	209.66	143.51	14.54	17.66	1061.65	572.46	511.72	928.09	926.87	1.43	1.46	1.30	1.41	1.41	24.13	13.01	11.63	21.09	21.07
Inst40T1.1.G27	103.79	71.85	7.41	3.42	938.34	369.11	433.96	929.12	934.13	1.35	1.30	1.25	1.37	1.35	17.70	6.96	8.19	17.53	17.63
Inst40T1.2.G27	125.32	123.63	8.36	6.44	1077.09	761.51	700.57	1047.37	1050.54	1.55	1.50	1.46	1.54	1.54	25.65	18.13	16.68	24.94	25.01
Inst40T1.3.G27	90.45	8.55	4.62	4.51	1964.49	585.03	1964.21	1964.49	1964.49	2.31	2.26	2.31	2.31	2.31	41.80	12.45	41.79	41.80	41.80
Inst50T1.1.G27	201.13	116.83	5.32	5.07	1668.75	1101.91	1282.25	1666.11	1666.11	1.63	1.61	1.57	1.63	1.63	38.81	25.63	29.82	38.75	38.75
Inst50T1.2.G27	144.82	15.82	26.13	9.05	1595.13	921.95	1580.12	1452.16	1595.13	2.11	2.09	2.11	2.11	2.11	24.54	14.18	24.31	22.34	24.54
Inst50T1.3.G27	323.86	280.40	13.92	11.78	646.32	356.36	286.72	626.27	628.84	1.32	1.29	1.21	1.32	1.32	10.77	5.94	4.78	10.44	10.48
Inst60T1.1.G27	868.11	153.10	8.05	7.50	1857.14	720.11	1478.36	1727.69	1847.02	1.34	1.39	1.33	1.34	1.34	26.16	10.14	20.82	24.33	26.01
Inst60T1.2.G27	1376.86	486.97	249.18	201.91	1727.42	1399.77	1726.98	1727.08	1727.06	4.29	4.30	4.29	4.29	4.29	22.73	18.42	22.72	22.72	22.72
Inst60T1.3.G27	171.24	59.83	97.63	28.94	1535.01	1250.06	1526.13	1510.48	1534.80	2.69	2.68	2.67	2.70	2.68	22.91	18.66	22.78	22.54	22.91
Inst70T1.1.G27	219.90	39.47	21.93	21.68	2239.87	1969.32	2227.25	2239.86	2239.86	3.85	3.84	3.84	3.85	3.85	32.00	28.13	31.82	32.00	32.00
Inst70T1.2.G27	440.05	172.16	117.42	225.79	2339.09	2082.79	2331.38	2335.01	2311.72	3.29	3.27	3.29	3.30	3.29	28.88	25.71	28.78	28.83	28.54
Inst70T1.3.G27	1100.74	357.68	97.24	182.36	2362.96	1235.85	1847.60	1717.27	1803.23	3.48	3.47	3.41	3.48	3.39	35.27	18.45	27.58	25.63	26.91
Inst20T1.1.G24	50.12	45.74	3.84	1.47	500.37	318.56	297.39	499.68	500.37	1.57	1.53	1.51	1.57	1.57	14.72	9.37	8.75	14.70	14.72
Inst20T1.2.G24	153.24	11.78	10.20	5.40	2256.61	1032.60	2233.63	2235.32	2256.61	2.66	2.69	2.64	2.65	2.66	60.99	27.91	60.37	60.41	60.99
Inst20T1.3.G24	185.55	20.53	10.00	7.88	2205.99	1361.81	2187.37	2116.00	2205.99	4.35	4.26	4.33	4.37	4.35	64.88	40.05	64.33	62.24	64.88
Inst30T1.1.G24	173.10	95.25	19.81	14.56	816.46	488.29	519.50	771.86	773.31	1.40	1.38	1.32	1.38	1.38	16.01	9.57	10.19	15.13	15.16
Inst30T1.2.G24	143.07	6.29	61.50	5.52	2832.25	1172.83	2822.48	2670.32	2832.25	1.78	1.82	1.77	1.79	1.78	57.80	23.94	57.60	54.50	57.80
Inst30T1.3.G24	110.00	12.93	5.16	4.12	2082.22	798.95	2004.47	2081.65	2082.22	3.01	3.06	2.98	3.02	3.01	54.80	21.03	52.75	54.78	54.80
Inst40T1.1.G24	175.31	5.69	17.39	5.90	2364.74	1586.70	2364.74	2361.15	2364.74	1.55	1.56	1.55	1.56	1.55	41.49	27.84	41.49	41.42	41.49
Inst40T1.2.G24	163.02	164.07	45.13	37.47	499.41	218.26	187.85	369.47	371.65	1.30	1.24	1.20	1.26	1.26	7.80	3.41	2.94	5.77	5.81
Inst40T1.3.G24	419.38	52.24	70.09	30.08	2008.03	1281.24	1972.79	2004.07	2007.92	3.62	3.65	3.62	3.62	3.62	35.23	22.48	34.61	35.16	35.23
Inst50T1.1.G24	309.10	288.27	22.65	16.13	322.24	198.05	169.16	263.89	279.62	1.34	1.38	1.24	1.32	1.31	5.86	3.60	3.08	4.80	5.08
Inst50T1.2.G24	630.35	33.24	209.46	43.62	2606.13	2232.98	2606.13	2557.33	2606.13	4.60	4.62	4.60	4.60	4.60	37.23	31.90	37.23	36.53	37.23
Inst50T1.3.G24	473.24	268.59	63.52	16.33	1403.93	899.57	1356.18	1383.18	1395.98	2.05	2.05	2.03	2.07	2.05	20.06	12.85	19.37	19.76	19.94
Inst60T1.1.G24	1073.37	1296.69	15.79	12.28	797.37	498.82	537.73	767.74	786.82	1.31	1.31	1.25	1.31	1.31	9.84	6.16	6.64	9.48	9.71
Inst60T1.2.G24	197.50	22.96	16.41	10.62	3239.33	2246.53	3152.07	3239.32	3239.33	1.74	1.74	1.73	1.74	1.74	46.95	32.56	45.68	46.95	46.95
Inst60T1.3.G24	259.27	13.15	86.91	13.37	2476.19	1921.24	2476.19	2437.95	2476.19	1.44	1.46	1.44	1.44	1.44	30.57	23.72	30.57	30.10	30.57
Inst70T1.1.G24	859.45	684.51	57.21	33.92	973.29	802.11	813.72	961.74	963.09	1.31	1.30	1.24	1.30	1.30	11.87	9.78	9.92	11.73	11.75
Inst70T1.2.G24	1273.10	53.68	125.31	35.57	2720.66	1894.23	2693.98	2700.54	2720.66	2.94	2.95	2.93	2.94	2.94	28.94	20.15	28.66	28.73	28.94
Inst70T1.3.G24	981.57	69.70	4536.10	38.24	2342.35	1506.07	2332.73	1177.07	2342.35	2.71	2.72	2.71	2.47	2.71	21.69	13.95	21.60	10.90	21.69

In Table 3 we depict the results of sequential VND, of sequential VNS and of the best solution found with basic LS. VNS was run three times, and the results correspond to an average of the three results obtained. VND and VNS average solution time increases 1.3 and 5.9 times, respectively, when compared with the time to find the best LS solution. VND improves total tardiness in 2 instances (improvement of 0.88 hours in Inst60T1.2.G21 and of 6.65 hours in Inst70T1.3G27) and improves total deviation in 44 instances. VNS improves total tardiness in 4 instances (with improvements of 1 hour, 0.66 hours, 33.36 hours and 0.13 hours in instances Inst60T1.2G21, Inst70T1.1.G27, Inst70T1.3G27 and Inst70T1.3.G24 respectively) and improves total deviation in 42 instances. VND improves the global objective in 45 instances and VNS in 43 instances, but the average improvement (considering only the instances with improvement) with VNS is almost 4.5 times greater than the average improvement with VND. VND and VNS improve the global objective of the best LS solution in 0.22% and 1.35%, respectively. Note that this measure considers all the 54 instances.

Table 3 – Best basic local search, VND and VNS results

	Solution time (seconds)			Total deviation (hours)			Average number of lots per component			Average deviation per product		
Instance	Basic local search	VND	VNS	Basic local search	VND	VNS	Basic local search	VND	VNS	Basic local search	VND	VNS
Inst20T1.1.G21	0.27	0.36	0.63	7.79	6.13	2.44	1.11	1.11	1.06	0.97	0.77	0.31
Inst20T1.2.G21	0.14	0.47	1.44	695.97	683.77	687.84	1.60	1.52	1.55	77.33	75.97	76.43
Inst20T1.3.G21	0.09	0.23	0.95	456.27	456.27	456.27	2.59	2.59	2.59	50.70	50.70	50.70
Inst30T1.1.G21	3.89	10.06	20.04	1091.35	907.99	955.12	2.34	2.24	2.26	60.63	50.44	53.06
Inst30T1.2.G21	1.42	4.06	14.89	328.25	181.20	178.55	1.32	1.29	1.23	16.41	9.06	8.93
Inst30T1.3.G21	2.65	4.13	11.02	412.24	410.31	412.24	2.13	2.12	2.13	20.61	20.52	20.61
Inst40T1.1.G21	1.75	2.86	11.33	360.48	360.11	358.91	1.28	1.28	1.28	18.02	18.01	17.95
Inst40T1.2.G21	3.96	7.43	30.44	850.16	844.12	848.15	1.65	1.62	1.64	35.42	35.17	35.34
Inst40T1.3.G21	4.91	6.41	28.04	157.29	154.09	120.09	1.26	1.24	1.18	5.83	5.71	4.45
Inst50T1.1.G21	20.26	22.37	32.85	173.63	173.25	170.13	1.19	1.19	1.17	5.99	5.97	5.87
Inst50T1.2.G21	8.60	12.64	27.47	592.59	506.79	535.39	1.43	1.42	1.42	19.75	16.89	17.85
Inst50T1.3.G21	105.38	34.15	102.39	275.64	267.12	246.86	1.14	1.16	1.15	8.35	8.09	7.48
Inst60T1.1.G21	20.44	25.32	50.61	760.65	711.99	747.63	1.36	1.31	1.33	29.26	27.38	28.75
Inst60T1.2.G21	3.31	9.22	31.91	631.99	502.98	542.47	1.43	1.43	1.42	21.07	16.77	18.08
Inst60T1.3.G21	16.26	19.86	51.58	207.63	174.00	184.00	1.48	1.47	1.46	7.16	6.00	6.34
Inst70T1.1.G21	9.61	17.08	54.03	423.69	408.92	418.75	2.74	2.70	2.72	14.12	13.63	13.96
Inst70T1.2.G21	33.87	37.16	80.52	113.14	97.35	72.72	1.16	1.16	1.09	3.43	2.95	2.20
Inst70T1.3.G21	23.29	52.23	174.76	160.52	143.29	159.91	1.14	1.16	1.18	4.12	3.67	4.10
Inst20T1.1.G27	70.78	93.29	417.30	262.17	248.73	206.65	1.35	1.33	1.31	8.46	8.02	6.67
Inst20T1.2.G27	43.15	57.19	295.58	519.89	515.96	519.89	2.12	2.10	2.12	16.25	16.12	16.25
Inst20T1.3.G27	11.54	18.69	96.26	397.28	392.00	391.21	1.43	1.42	1.43	13.70	13.52	13.49
Inst30T1.1.G27	66.47	81.49	722.77	215.23	282.73	150.68	1.35	1.49	1.30	5.25	6.90	3.68
Inst30T1.2.G27	103.88	146.34	782.15	1584.96	1576.08	1580.71	3.57	3.55	3.57	41.71	41.48	41.60
Inst30T1.3.G27	143.51	232.88	708.05	511.72	571.84	564.55	1.30	1.45	1.44	11.63	13.00	12.83
Inst40T1.1.G27	103.79	151.52	942.35	369.11	360.55	393.43	1.30	1.30	1.36	6.96	6.80	7.42
Inst40T1.2.G27	123.63	170.56	1694.33	700.57	742.43	722.65	1.46	1.48	1.47	16.68	17.68	17.21
Inst40T1.3.G27	90.45	244.37	2132.32	585.03	542.08	549.79	2.26	2.25	2.25	12.45	11.53	11.70
Inst50T1.1.G27	201.13	252.28	1661.01	1101.91	1089.49	1101.91	1.61	1.60	1.61	25.63	25.34	25.63
Inst50T1.2.G27	144.82	210.87	1499.47	921.95	919.65	921.18	2.09	2.09	2.09	14.18	14.15	14.17
Inst50T1.3.G27	280.40	401.75	1258.70	286.72	335.02	285.21	1.21	1.28	1.24	4.78	5.58	4.75
Inst60T1.1.G27	868.11	1613.57	3865.67	720.11	626.08	659.01	1.39	1.37	1.38	10.14	8.82	9.28
Inst60T1.2.G27	1376.86	1936.93	7054.15	1399.77	1383.10	1394.00	4.30	4.28	4.30	18.42	18.20	18.34
Inst60T1.3.G27	171.24	435.21	3397.20	1250.06	1203.43	1218.96	2.68	2.70	2.69	18.66	17.96	18.19
Inst70T1.1.G27	219.90	377.75	2863.70	1969.32	1954.39	1979.39	3.84	3.84	3.81	28.13	27.92	28.28
Inst70T1.2.G27	440.05	2277.10	6650.74	2082.79	1910.55	2077.18	3.27	3.30	3.28	25.71	23.59	25.64
Inst70T1.3.G27	1100.74	1731.82	7202.54	235.85	1042.37	1038.58	3.47	3.28	3.09	3.52	15.56	15.50
Inst20T1.1.G24	45.74	76.27	454.16	297.39	310.00	298.98	1.51	1.51	1.53	8.75	9.12	8.79
Inst20T1.2.G24	153.24	219.12	1422.12	1032.60	970.48	993.93	2.69	2.63	2.65	27.91	26.23	26.86
Inst20T1.3.G24	185.55	299.61	1957.34	1361.81	1320.84	1361.81	4.26	4.26	4.26	40.05	38.85	40.05
Inst30T1.1.G24	173.10	226.29	1608.08	488.29	475.27	476.39	1.38	1.36	1.36	9.57	9.32	9.34
Inst30T1.2.G24	143.07	240.13	1871.80	1172.83	1131.31	1172.83	1.82	1.76	1.82	23.94	23.09	23.94
Inst30T1.3.G24	110.00	159.67	693.92	798.95	781.61	798.95	3.06	3.04	3.06	21.03	20.57	21.03
Inst40T1.1.G24	175.31	279.88	2324.63	1586.70	1525.96	1531.06	1.56	1.55	1.55	27.84	26.77	26.86
Inst40T1.2.G24	164.07	238.87	937.52	187.85	208.72	193.61	1.20	1.23	1.23	2.94	3.26	3.03
Inst40T1.3.G24	419.38	594.80	4048.48	1281.24	1250.78	1260.92	3.65	3.62	3.63	22.48	21.94	22.12
Inst50T1.1.G24	288.27	422.35	1402.81	169.16	180.85	176.57	1.24	1.32	1.31	3.08	3.29	3.21
Inst50T1.2.G24	630.35	1411.16	7209.70	2232.98	2127.91	2174.05	4.62	4.63	4.63	31.90	30.40	31.06
Inst50T1.3.G24	473.24	655.57	1886.94	899.57	893.16	900.35	2.05	2.03	2.04	12.85	12.76	12.86
Inst60T1.1.G24	1073.37	1556.55	5635.47	498.82	457.90	468.77	1.31	1.24	1.27	6.16	5.65	5.79
Inst60T1.2.G24	197.50	395.98	3366.48	2246.53	2211.74	2246.53	1.74	1.76	1.74	32.56	32.05	32.56
Inst60T1.3.G24	259.27	565.45	3954.17	1921.24	1866.57	1903.02	1.46	1.46	1.46	23.72	23.04	23.49
Inst70T1.1.G24	859.45	1164.45	3764.07	802.11	798.97	794.38	1.30	1.29	1.27	9.78	9.74	9.69
Inst70T1.2.G24	1273.10	1779.32	6487.63	1894.23	1855.63	1894.23	2.95	2.95	2.95	20.15	19.74	20.15
Inst70T1.3.G24	4536.10	1516.27	7202.60	1177.07	1471.45	1494.20	2.47	2.71	2.66	10.90	13.62	13.84

Conclusion

In this paper we presented several approximate methods for a real-world lot splitting and scheduling problem existent in a textile company. As far as we are aware, this is the first time this problem is considered in the literature. We are dealing with a NP-hard problem for which is not easy to find optimal solutions.

As expected, the basic local search algorithms and the metaheuristics improve the best initial solutions for the majority of the instances tested at the expense of spending more computational time. A great amount of the instances tested are greater in size compared to the real ones. The computational experiments show that feasible solutions can be found quickly even for high machine utilizations. We believe that the developed work is of major interest in the context of the real problem and can effectively aid the decision maker in developing production plans.

Acknowledgements

This work was supported by the Portuguese Science and Technology Foundation through the doctoral Grant SFRH / BD / 38582 / 2007 for Carina Pimentel.

References

Allahverdi, A., Ng, C., Cheng, T. and Kovalyov, M. (2008), "A survey of scheduling problems with setup times or costs", *European Journal of Operational Research,* Vol.187, pp. 985-1032.

Anderson, E., Glass, C. and Potts, C. (1997), "Machine scheduling", in Aarts, E. & Lenstra, J. (Ed.), *Local Search in Combinatorial Optimization,* John Wiley and Sons Ltd, Chichester, pp. 361-414.

Blum, C. and Roli, A. (2003), "Metaheuristics in combinatorial optimization: overview and conceptual comparison", *ACM Computing Surveys,* Vol.35, pp. 268-308.

Cheng, T. and Sin, C. (1990), "A state-of-the-art review of parallel-machine scheduling research", *European Journal of Operational Research,* Vol.47, pp. 271-292.

Hansen, P. and Mladenovic, N. (2001), "Variable neighborhood search: principles and applications", *European Journal of Operational Research,* Vol.130, pp. 449-467.

Hertz, A. and Mittaz, M. (2001), "A variable neighborhood descent algorithm for the undirected capacitated arc routing problem", *Transportation Science,* Vol.35**,** No.4, pp. 425-434.

ILOG (2007), CPLEX 11.0 User's Manual.

Mokotoff, E. (2001), "Parallel machine scheduling problems: a survey", *Asia-Pacific Journal of Operational Research,* Vol.18, pp. 193-242.

Pimentel, C., Alvelos, F., Duarte, A. and Carvalho, J.(2010a) "Exact and heuristic approaches for lot splitting and scheduling on identical parallel machines", *International Journal of Manufacturing Technology and Management* (accepted for publication).

Pimentel, C., Alvelos, F., Duarte, A. and Carvalho, J. (2010b) "A fast heuristic for a lot splitting and scheduling problem of a textile industry", University of Minho, Working paper.

Sheen, G. and Liao, L. (2007), "Scheduling machine-dependent jobs to minimize lateness on machines with identical speed under availability constraints", *Computers and Operations Research,* Vol.34, pp. 2266-2278.

Tahar, D., Yalaoui, F., Chu, C. and Amodeo, L. (2006), "A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times", *International Journal of Production Economics,* Vol.99, pp. 63-73.

Xhafa, F. and Abraham, A. (2008) "Metaheuristics for scheduling in industrial and manufacturing applications", in Kacprzyk, J. (Ed.), *Studies in Computational Intelligence,* Springer.

Xing, W. and Zhang, J. (2000), "Parallel machine scheduling with splitting jobs", *Discrete Applied Mathematics,* Vol.103, pp. 259-269.

Zhu, X. and Wilhelm, W. (2006), "Scheduling and lot sizing with sequence-dependent setup: a literature review", *IIE Transactions,* Vol.38, No.11, pp. 987-1007.