

Tabu Search to Solve the Synchronized and Integrated Two-Level Lot Sizing and Scheduling Problem

Claudio Fabiano Motta Toledo
Institute of Mathematics and Computer
Science - USP
Av. Trabalhador são-carlense, 400,
13566-590
55 (16) 3373-9700
claudio@icmc.usp.br

Márcio da Silva Arantes
Dept. of Computer Science –
University of Lavras
CP3037 – 37200-000
Lavras, MG
55 (35) 3829-1545
marcio@comp.ufla.br

Paulo Morelato França
Dept. of Mathematics, Statistics and
Computation - UNESP
CP 266 – 19060-900
P. Prudente, SP, Brazil
55 (18) 3229-5385
paulo.morelato@fct.unesp.br

ABSTRACT

This paper proposes a tabu search approach to solve the Synchronized and Integrated Two-Level Lot Sizing and Scheduling Problem (SITLSP). It is a real-world problem, often found in soft drink companies, where the production process has two integrated levels with decisions concerning raw material storage and soft drink bottling. Lot sizing and scheduling of raw materials in tanks and products in bottling lines must be simultaneously determined. Real data provided by a soft drink company is used to make comparisons with a previous genetic algorithm. Computational results have demonstrated that tabu search outperformed genetic algorithm in all instances.

Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control methods, and Search – *heuristic methods, scheduling*

General Terms: Algorithms, Management, Measurement, Performance, Experimentation, Verification.

Keywords

Tabu search, genetic algorithm, scheduling, real-world applications, industrial applications

1. INTRODUCTION

This paper presents a tabu search algorithm to solve the Synchronized and Integrated Two-Level Lot Sizing and Scheduling Problem (SITLSP). This problem is usually found in soft drink companies, where lot sizing and scheduling decision have to be simultaneously taken in a two-level production process. The first level consists of tanks storing raw materials that spread to bottling lines. Decisions about lot size and scheduling for raw materials stored in tanks must be made. At the same time, the lot size of each demanded product and its corresponding production scheduling in the available lines must be determined in the second level.

A literature review about lot sizing and scheduling problems can be found in [11] where models are presented for different lot sizing problems.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'11, July 12–16, 2011, Dublin, Ireland.

Copyright 2011 ACM 978-1-4503-0557-0/11/07...\$10.00.

In [18] a review for capacitated lot sizing problems with discussions about formulation and complexity issues is presented. In [16] an overview on modeling for industrial extensions of single-level dynamic lot sizing problems is provided. Operational aspects like setup, production process, inventory, demand, and time horizon are taken into account on modeling. A review on metaheuristics applied to solve lot sizing problem is reported by [17]. Solution representation, evaluation functions, neighborhoods and operators are discussed as well as computational experiments. Models and algorithms are reviewed by [22] for the dynamic demand and coordinate lot-sizing problems. This class of problem includes issues as single and multiple items, coordinated and uncoordinated setup structures, uncapacitated and capacitated problems.

A mixed integer programming (MIP) model for a lot sizing and scheduling problem in foundry industries is proposed in [3]. It is a multi-item, single machine problem with sequence-dependent setup costs and times. A fix-and-relax method using local search procedures as a descendent heuristic, diminishing neighborhood and simulated annealing is introduced. In [4] rolling-horizon and fix-and-relax heuristics to solve the identical parallel machine lot sizing and scheduling problem with sequence dependent set up costs are proposed. The fix-and-relax heuristic outperforms rolling-horizon and its results are close to the instances' lower bounds. The recent papers [10,20] present MIP models for lot sizing and scheduling in the presence of setup times for problems found in electrofused grain and animal nutrition industries. A lot sizing and scheduling problem from a manufacturing plant for animal feed compounds is studied in [25]. Model formulations and fix-and-relax heuristics are proposed.

The MIP model for the SITLSP was first presented in [24] and it can be solved using commercial solvers, but just for small-sized instances. A lot sizing problem from a drink manufacturer with a single line (canning line) and sequence-independent setup times for products is presented in [9]. The author introduces a MIP model and several heuristics to solve it. A problem similar to SITLSP, where each bottling line has a dedicated tank and each tank can be filled with all liquid flavors needed by this line, is proposed in [12]. Relax-and-fix heuristics were tested by authors using instances problems based on real data from a leading market soft drink company. Computational results showed that the relax-and-fix approach yielded better production plans compared to those currently used by the company. A MIP model for a small-scale soft drink problem is presented in [13]. It is simpler than the MIP models presented in [12] and [24] once that there is just one production line. It is also

unnecessary to schedule raw material in tanks, so there are not setup times and costs at tank level. This situation is more usual in small plants.

In [23] a multi-population genetic algorithm (GA) to solve SITLSP instances is introduced. This GA uses a population hierarchically structured in ternary trees. The authors proposed a tailor-made representation of solution as GA individuals. That same solution representation is used in the TS approach introduced in the present paper.

This paper is organized as follows. The SITLP is described in Section II and the TS approach is presented in Section III. Computational tests are reported in Section IV and conclusions follow in Section V.

2. SYNCHRONIZED AND INTEGRATED TWO-LEVEL-LOT SIZING AND SCHEDULING PROBLEM - SITLSP

The problem studied in this paper is frequently found in industrial settings mainly soft drink companies. The studies of this work were conducted based on data provided by a leading market soft drink company in Brazil. There are more than 800 beverage companies in Brazil producing wine, juice, beer, mineral water and soft drinks. The Brazilian soft drink market is the third largest in the world [1] and soft drink consumption has increased in the last years with more than 14 billions of liters annually consumed. Fig. 1 illustrates a typical production process as considered in this paper.

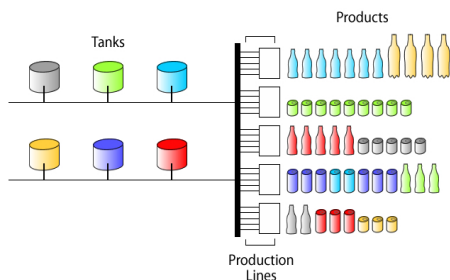


Figure 1. The two-level production process.

There are two interdependent levels with decisions about raw material storage and soft drink bottling. In the upper level, capacitated mixing tanks are used to prepare and store raw materials which are pumped to the lower level with non-identical parallel production lines. Tanks are only filled up again when empty and two different raw materials cannot be stored at the same time in the same tank. It is necessary to clean and fill up a tank and no raw material can be pumped from this tank to a production line during this time. Therefore, a sequence-dependent setup time and setup costs take place.

At the lower level, raw materials from tanks are bottled in several bottle types of different sizes. The final product in each line is both defined by the flavor of the soft drink and the bottle type. If a line has two different products switched, a sequence-dependent setup time must be also respected. Each line produces a set of products and the same product can be processed in more than one line. The processing time of each product depends on the bottle types and the lines used to produce the final product.

Demands usually must be weekly met and the excessive number of final products leads to inventory costs. There are also inventory

costs for the storage of raw material in tanks during consecutive time periods. A production cost is incurred for each unit of product and raw material. The sequence-dependent setup costs are usually proportional to the sequence-dependent setup times in lines and tanks, respectively. Synchronization is necessary once that a product cannot be produced by lines without its corresponding raw material having been scheduled in tanks. The most challenging aspect of the problem is the integration of all these issues in a synchronized two-level problem.

The mathematical model that describes this problem was first defined in [24]. It is a MIP model that encompasses two Capacitated Lot Sizing Problems (CLSP), one in the upper level and another in the lower level. These problems are synchronized and integrated by a set of constraints. In [6] the CLSP is proven to be NP-hard. The SITLSP is a multi-item problem because several raw materials and products in each level of the SITLSP as to be managed. As the multi-item CLSP is strongly NP-hard [8], it is expected that real-world SITLSP problems are only suitable to be solved by approximate methods. Moreover, in most applications raw materials and products must consider sequence-dependent setup times and costs. In [21] it is showed that even finding feasible solutions for CLSP with setup times is a NP-complete problem.

3. TABU SEARCH ALGORITHM

Tabu search (TS) is a known metaheuristic that uses a local search technique coupled with a short-term memory structure to avoid visiting the same solutions or neighborhoods several times [14]. It has been successfully used to solve several lot sizing and scheduling problems. TS is used in [15] to solve a production planning problem with setups (time and cost) and multiple products, resources and periods. The potential solutions (neighbors) are evaluated solving LP problems whose information allows defining a rank of neighbors. This rank is used to evaluate different strategy approaches to explore neighborhoods. A TS approach [2] is employed to solve a production planning problem in a flexible production system. A lot sizing model with backorder and a set of random test problems are presented. The TS outperforms a random sequencing procedure within a short CPU time. A multilevel resource-constrained lot-sizing problem with setup and lead times is solved by a TS approach [5] which uses backward-forward movements to adjust solutions to feasibility. TS is also used by [7] to solve the job shop scheduling problem taking into account lot streaming. The heuristic is used in task scheduling, and moves are defined to generate new schedules from an initial one. The moves that can allow returning to previous schedules are kept tabu for several iterations.

3.1 Representation, decoding and evaluation of solution

The representation of solution presented is close to the one in [19] that uses a two-dimensional matrix with assignment rules for a multi-level proportional lot sizing and scheduling problem with multiple machines. Fig. 2 shows each entry (t, n) , $t \in T$ and $n \in N$ of the two-dimensional matrix with T (number of periods) rows and N columns.

$P_{t,n} \mid LS_{t,n}$
$\alpha_1, \dots, \alpha_k$
β_1, \dots, β_k

Figure 2. Entry (t,n) for representation of solution

- $P_{t,n}$: product in position n to be produced in period t .
- $LS_{t,n}$: lot size of product $P_{t,n}$.
- $SL_{t,n}$: sequence of lines where $P_{t,n}$ can be produced.

$$SL_{t,n} = (\alpha_1, \dots, \alpha_k).$$

- $STk_{t,n}$: sequence of tanks where the raw material of $P_{t,n}$ can be stored.

$$STk_{t,n} = (\beta_1, \dots, \beta_k).$$

In this representation of solution, the demand D_{it} of some product i in period t is divided into several lots ($LS_{t,n}$) and randomly distributed among entries in periods $t, t-1, t-2, \dots, 1$. Backorders are not allowed. Sequences $SL_{t,n}$ and $STk_{t,n}$ are randomly generated with length k . In sequence $SL_{t,n} = (\alpha_1, \dots, \alpha_k)$ with $\alpha_i \in \{1, \dots, L\}$, α_i is a line number and L is the number of lines. This sequence is created taking into account only lines where product $P_{t,n}$ can be produced. In sequence $STk_{t,n} = (\beta_1, \dots, \beta_k)$ with $\beta_i \in \{1, 2, \dots, 2\bar{L}\}$, β_i keeps information about tank number and how the raw material will be stored in this tank. It is necessary taking from β_i the tank j where raw material is stored:

$$j = \begin{cases} \beta_i & 1 \leq \beta_i \leq \bar{L} \\ \beta_i - \bar{L} & \bar{L} < \beta_i \leq 2\bar{L} \end{cases}$$

The parameter \bar{L} is the number of tanks. If $1 \leq \beta_i \leq \bar{L}$, the tank $j = \beta_i$ will be occupied after the raw material previously stored has been used. This forces the method to find solutions where there is a partial use of tank capacity. If $\bar{L} < \beta_i \leq 2\bar{L}$, the tank $j = \beta_i - \bar{L}$ will be immediately occupied. This forces the method to find solutions where the tank capacity is completely filled.

There are exceptions for these rules. If tank j stores a raw material different from the raw material used by product $P_{t,n}$, it will be necessarily occupied after the previous raw material stored has been used. The same idea is applied when tank j is completely full. On the other hand, if the selected tank j is empty it will be immediately occupied.

Figure 3 illustrates two possible representations of solutions. Suppose two products (P_1 and P_2) with 100 units of demand to be satisfied on periods T_1 and T_2 . Let's assume that each product uses different raw materials. There are also two lines and two tanks available to produce and store raw materials and products, respectively.

	Solution 1			Solution 2			
T_1	$P_2 \mid 50$ 2221 1124	$P_1 \mid 100$ 1221 4432	$P_2 \mid 50$ 2211 3233	$P_2 \mid 40$ 2112 2341	$P_1 \mid 100$ 1121 3121	$P_2 \mid 60$ 2221 4421	$P_1 \mid 45$ 2121 3421
T_2	$P_2 \mid 100$ 1112 2331	$P_1 \mid 100$ 2212 1342		$P_2 \mid 30$ 2221 4124	$P_1 \mid 55$ 2122 3323	$P_2 \mid 70$ 1212 4421	

Figure 3. Representing solutions for SITLSP

Figure 3 shows three entries in the first matrix line of solution 1 and four entries in the first matrix line of solution 2. This first line encodes information to define lot sizing and scheduling for tanks and lines within the first time period (T_1). A decoding procedure to do that will be explained next.

The first entry in solution 1 has 50 units ($LS_{1,1}$) of product P_2 with $SL_{1,1} = (2,2,2,1)$ as sequence of lines and $STk_{1,1} = (1,1,2,4)$ as sequence of tanks. The sequence of lines ($SL_{t,n}$) and tanks ($STk_{t,n}$) can repeat values, where $\alpha_i \in \{1,2\}$ and $\beta_i \in \{1,2,3,4\}$ for $L = \bar{L} = 2$ and $k = 4$ (sequence length).

The demands are distributed in their respective periods in solution 1, but the demand of P_2 in T_1 is split between two entries with 50 units each one. In solution 2, part of the P_1 demand in T_2 is produced in T_1 , where it is shown an entry with 45 units of P_1 .

A procedure decodes the information stored in each representation of solution into a problem solution. In Fig. 4 a pseudo code explains how these representations of solution become a problem solution.

```

Method DecodingSolRepr(SolutionRepres)
  Entryt,n ← getInitialEntry(SolutionRepres);
  LSt,n ← getDemand(Entryt,n);
  again ← true;
  begin
    While (again) Do
      ( $\alpha_i, \beta_i$ ) ← selectPairOfLineAndTank(Entryt,n);
      While ( $\alpha_i, \beta_i \neq \emptyset$ ) Do
        auxLSt,n = insertLotSize(LSt,n,  $\alpha_i, \beta_i$ );
        LSt,n = LSt,n - auxLSt,n;
        If (LSt,n == 0) then ( $\alpha_i, \beta_i$ ) =  $\emptyset$ 
        Else ( $\alpha_i, \beta_i$ ) ← selectPairOfLineAndTank(Entryt,n);
      endWhile;
      Entryt,n ← getNextEntry(SolutionRepres);
      If (Entryt,n ==  $\emptyset$ ) then again = false;
    endWhile.
  end.

```

Figure 4. Pseudo code for decoding a representation of solution

The method `getInitialEntry(SolutionRepres)` returns the initial entry whose data will be used to generate a solution. The initial entry is the first entry in the last period of solution. The method starts with this first entry and selects the next entries until it reaches the last one at the first period. This backward process aims to postpone setups and processing time. However, there is no guarantee that all demands will be produced at the end. The lot size ($LS_{t,n}$) of each entry will be scheduled using pairs (α_i, β_i) selected from the sequence of values in $SL_{t,n}$ and $STk_{t,n}$ by method `selectPairOfLineAndTank(Entryt,n)`.

The method `insertLotSize(LSt,n, α_i, β_i)` evaluates if the selected pair (α_i, β_i) returns a line and a tank with available capacity to produce $LS_{t,n}$ or at least a fraction of $LS_{t,n}$ ($auxLS_{t,n}$). A next pair ($\alpha_{i+1}, \beta_{i+1}$) of lines and tanks is selected again from $SL_{t,n}$ and $STk_{t,n}$ while $LS_{t,n} > 0$ or ($\alpha_i, \beta_i \neq \emptyset$).

The decoding procedure returns a lot size and schedule (problem solution) simultaneously for products and raw materials in lines and tanks. This solution is evaluated using the objective function of the problem that is determined adding up all costs: production, setup and inventory costs for products and raw materials in lines and tanks, respectively. If some demands are not satisfied, a high penalty cost per unit is also added. The best solution is the one whose decoding process determines a final schedule with a minimum cost value. Details about decoding process and objective function of SITLSP can be found in [24] and [23].

3.2 Pseudo code and moves

Fig. 5 shows the pseudo code for the proposed TS. The method is executed until a time limit has been reached. A current solution S is changed by method *executeNeighborhoodMoves(S)* that returns the best neighbor (*bestNeighbor*) and the respective move that led to this neighbor (*bestMove*). The current solution is updated by *bestNeighbor* and the *bestMove* is inserted in tabu list. The best solution S^* may also be updated next. A dynamic tabu list is used, so its length changes during the TS execution after a certain number of iterations. In this way, if the size of tabu list is reduced, the older moves become free earlier. Otherwise, they will remain longer as tabu. The tabu list length is determined as a random integer value $tabuListLength \in \{1, \dots, numberOfMoves-1\}$.

```

Tabu Search 0
  tabuList  $\leftarrow \emptyset$ 
  Initialize(S)
   $S^* \leftarrow S$ 
   $f(S^*) \leftarrow f(S)$ 
  count  $\leftarrow 0$ ;
  while (execution time was not reached) do
    ( $S\_bestNeighbor, bestMove$ )  $\leftarrow executeNeighborhoodMoves(S)$ 
     $S \leftarrow S\_bestNeighbor$ ;
    insertTabuMove (TabuList, mBest)
    If ( $f(S) < f(S^*)$ )
       $S^* \leftarrow S$ 
    count  $\leftarrow$  count + 1;
    If (count > maxCount)
      tabuListLength  $\in \{1, \dots, numberOfMoves-1\}$ 
      updateTabuListLength (tabuList, tamanhoListaTabu)
      count  $\leftarrow 0$ ;
  endWhile
  return ( $S^*, f(S^*)$ )
endTabuSearch

```

Figure 5. Tabu search algorithm pseudo code

A total of five neighborhood moves are applied over the representation of solution described in section 3. The first move is a swap where two selected entries are changed (Fig. 6). The second move executes an insertion, where an entry is inserted into another position (Fig. 7). The third move takes two entries with the same product, select one of them to merge the lot sizes and the other to be removed (Fig. 8). The fourth move split the lot size in one entry. A new entry is created in such a way that it receives part of lot size, and the other piece is kept on the original entry. Next, the new entry is inserted into another position (Fig. 9). The fifth move reinitializes the sequence of tanks and lines in an entry (Fig. 10).

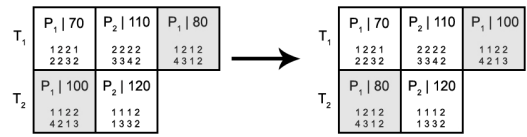


Figure 6. Swap move

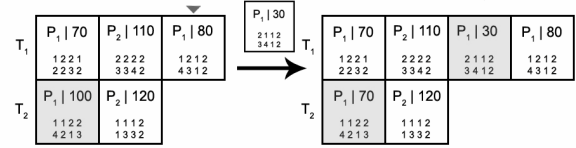


Figure 7. Insertion move

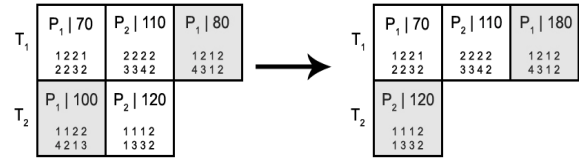


Figure 8. Merge move

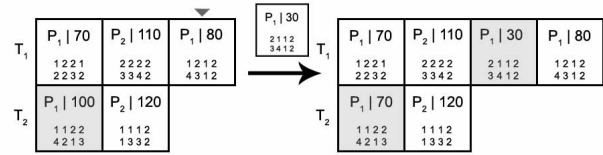


Figure 9. Split move

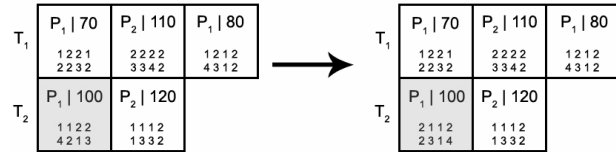


Figure 10. Reinitialize sequences move

These moves select entries randomly, but it is not allowed violating demand satisfaction. For instance, it is not possible to insert an entry on the second period if its lot has to fill demand in the first period. In this case, another entry is randomly chosen. The proposed moves will execute neighborhood exploration through *executeNeighborhoodMoves(S)*, where S is the current solution (Figure 11).

```

executeNeighborhoodMoves (S)
  for  $a \leftarrow 1$  to numberOfMoves do
     $m \leftarrow selectAvailableMove()$ 
    for  $b \leftarrow 1$  to maxIter do
       $S\_temp \leftarrow executeMove(m, S)$ 
      If  $f(S\_temp) < f(S)$ 
         $S\_aux \leftarrow S\_temp$ 
        moveAux  $\leftarrow m$ 
    endFor
  endFor
  return ( $S\_aux, moveAux$ )
endExecuteNeighborhoodMoves

```

Figure 11. Pseudo code for executeNeighborhoodMoves(S)

There is a main loop where a move m is defined by *selectAvailableMove()*. The move m is randomly selected among the available moves. Next, move m is applied over S until *maxIter* has been reached. At the end, the best solution is stored. The best solution found after *numberOfMoves* executions being reached is returned, as well as the move which has led to it.

4. COMPUTATIONAL RESULTS

Problem instances were defined from data provided by a leader market soft drink company. These data are based on production plans executed by company during several time periods. Table 1 presents the parameters used to characterize each instance.

Table 1. Parameters for instances

Inst.	Lines	Tanks	Products	Raw materials	Periods
A1	5	9	33	11	1
A2	6	9	49	14	2
A3	6	9	58	15	3
B1	6	10	52	19	1
B2	6	10	56	19	2
B3	6	10	65	21	3

Instances type A has 7 days in each production time period and the instances type B have 10 days in each time period. Lines and tanks are available 24 hour for production. The most complex type A instance (A3) has 3 periods (3 weeks) with demands for 58 products that need 15 different raw materials. There are 6 lines and 9 tanks responsible for satisfying product demands at the end of each period. The most complex B instance (B3) has also 3 time periods (30 days at all) with 65 products (21 raw materials), 6 lines and 10 tanks.

There are other relevant parameter values present in all instances. The setup time is 0.5 hour for all products adjusted to any line. The setup cost was evaluated as 3000 \$/u also for all products adjusted to any line. There are no setup times and setup costs if two products produced in sequence are the same. However, this situation is different on the tank level. It is spent 1 hour to setup the same raw material and 2 hours to setup different raw materials in any tank. Therefore, it is evaluated as 6,000 \$/u the cost to setup the same raw material and 12,000 \$/u to setup a different raw material in tank. The lower and upper limits for tank capacity are 1000 and 24000 liters, respectively. Demands for the instances ranges from 47 to 180000 units. Line processing times for products range from 50 to 2000 units/hour. The inventory and production costs for products and raw materials are 1\$/u. All cost values were estimated by the industry and they reflect a suitable tradeoff between the different terms of the SITLSP objective function.

Production plans determined by the soft drink company for these instances were estimated by the cost values previously defined. The company solution estimated cost is represented by column Ind in Table 2. The proposed TS algorithm ran 10 times with execution time of 1 hour per execution over each instance. The objective function value Z_{TS} in Table 2 represents the average value of the best solutions found by TS in each run.

TS was adjusted to execute 5 moves (swap, insert, split, merge and restart rules), where the selected (not tabu) move is applied to create 40 neighbors. The length of tabu list is adjusted to reach value 4.

The proposed TS approach is also compared with the multi-population GA proposed in [23], executed on the same instance set and with the same CPU time. Again, the Z_{GA} value in Table 2 means the average value of the best solutions found by GA in each one of the 10 runs. Relative deviations from the cost obtained by industry planners are displayed in Table 2 as Dev(%). Computational tests were executed using an Intel Core 2 duo processor with 2.66GHz and 2GB RAM.

Table 2. Solution values and deviations

Inst.	Ind	GA		TS	
	Z_{IND}	Z_{GA}	Dev(%)	Z_{TS}	Dev(%)
A1	1692	1666	-1.5	1651	-2.4
A2	3511	3405	-3.0	3337	-5.0
A3	5002	5199	3.9	4915	-1.7
B1	3378	3271	-3.2	3199	-5.3
B2	4278	4231	-1.1	4053	-5.3
B3	7943	8056	1.4	7640	-3.8

All results found by TS approach have improved the ones attained by industry planners. Moreover, TS have outperformed GA, the best approach known so far. Notice that GA has not been able to improve the industry performance in 2 out of 6 instances.

5. CONCLUSIONS

In this paper a tabu search algorithm to solve the Synchronized and Integrated Two-Level Lot Sizing and Scheduling Problem (SITLSP) is proposed. In this two-level production planning problem, lot-sizing and scheduling decisions involving parallel machines, capacity constraints and sequence-dependent setup costs and times have to be made in both levels. A solution is represented by a matrix which contains information regarding lot sizes and sequencing for both levels. The tabu search implementation uses a dynamic tabu list and 5 different moves. A total of six instances based on data provided by a soft drink company are evaluated. The proposed metaheuristics have been compared with results found by a previous GA and with the estimated cost of a production plan elaborated by planners of the soft drink company. Comparisons have shown that tabu search has outperformed the two other approaches in all of the 6 instances.

6. ACKNOWLEDGMENTS

This research was supported by CNPq.

7. REFERENCES

- [1] ABIR, *Brazilian Association of Soft Drink Industry*, Report available in: <http://www.abir.gov.br> (last visit in 15/05/2010).

- [2] Al-Fawzan, M. A. An algorithm for production planning in a flexible production system. *Computer and Industrial Engineering*, 48(4), 2005, 681-691.
- [3] Araujo, S. A., Arenales, M. N. and Clark, A. R. Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups. *Journal of Heuristics*, 13, 2007, 337-358.
- [4] Beraldi, P., Ghianib, G., Griecob, A., and Guerriero, E. Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs. *Computers & Operations Research*, 35, 2008, 3644– 3656.
- [5] Berretta, R. E., França, P. M. and Armentano, V. A. Metaheuristic approaches for the multilevel resource-constrained lot-sizing problem with setup and lead times. *Asia-Pacific Journal of Operational Research*, 22 (2), 2005, 261-286.
- [6] Bitran, G. R. and Yanasse, H. H. Computational complexity of the capacitated lot size problem. *Management Science*, 28 (10), 1982, 1174-1186.
- [7] Buscher, U. and Shen, L. An integrated tabu search algorithm for the lot streaming problem in job shops. *European Journal of Operation Research*, 199(2), 2009, 385-399.
- [8] Chen, W. H. and Thizy, J. M. Analysis of relaxations for the multi-item capacitated lot-sizing problem. *Annals of Operations Research*, 26, 1990, 29-72.
- [9] Clark, A. R. Hybrid heuristics for planning lot setups and sizes. *Computers & Industrial Engineering*, vol. 45, 2003, 545-562.
- [10] Clark, A., Morabito, R. and Toso, E. Production setup-sequencing and lot-sizing at an animal nutrition plant through ATSP subtour elimination and patching. *Journal of Scheduling*, 13 (2), 2009, 111-121.
- [11] Drexler, A. and Kimms, A. Lot-sizing and scheduling - survey and extensions. *European Journal of Operational Research*, 99, 1997, 221-235.
- [12] Ferreira, D., Morabito, R. and Rangel, S. Solution approaches for the soft drink integrated production lot sizing and scheduling problem. *European Journal of Operational Research*, 196, 2009, 697-706.
- [13] Ferreira, D., Morabito, R. and Rangel, S. Relax and fix heuristics to solve one-stage one-machine lot-scheduling models for small-scale soft drink plants. *Computers & Operations Research*, 37, 2010, 684-691.
- [14] Glover, and Laguna, M. *Tabu Search*, Kluwer, Norwell, MA, 1997.
- [15] Hung, Y. F., Chen, C. P., Shih, C. C. and Hung, M. H. Using tabu search with ranking candidate list to solve production planning problems with setups. *Computers & Industrial Engineering*, 45, 2003, 615-634.
- [16] Jan, R. and Degraeve, Z. *Modeling industrial lot sizing problems: A review*. ERIM Report Series Reference, No. ERS-2005-049-LIS. Available at SSRN: <http://ssrn.com/abstract=902728>, 2005.
- [17] Jans, R. and Degraeve, Z. Metaheuristics for dynamic lot sizing: A review and comparison of solution approaches. *European Journal of Operational Research*, 177, 2007, 1855-1875.
- [18] Karimi, B., Ghomi Fatemi, S. M. T., and Wilson, J.M. The capacitated lot sizing problem: a review of models and algorithms. *Omega*, 31, 2003, 365—378.
- [19] Kimms, A. A genetic algorithm for multi-level, multi-machine lot sizing and scheduling. *Computers & Operations Research* 26, 1999, 829-848.
- [20] Luche, J. R. D., Morabito, R. and Pureza, V. Combining process selection and lot sizing models for production scheduling of electrofused grains. *Asia-Pacific Journal of Operational Research* 26(3), 2009, 421-443.
- [21] Maes, J., Mc Clain, J. O. and Van Wassenhove, L. N. Multilevel capacitated lot sizing complexity and LP-based heuristics. *European Journal of Operational Research*, 53(2), 1991, 131-148.
- [22] Robinson, P., Narayananb, A., and Sahinc, F. Coordinated deterministic dynamic demand lot-sizing problem: A review of models and algorithms. *Omega*, 37, 2009, 3– 15.
- [23] Toledo, C. F. M., França, P. M., Kimms, A. and Morabito, R. A multi-population genetic algorithm approach to solve the synchronized and integrated two-level lot sizing and scheduling problem. *International Journal of Production Research*, 47, 2009, 3097-3119.
- [24] Toledo, C. F. M., França, P. M., Morabito, R. and Kimms, A. An optimization model for the synchronized and integrated two-level lot sizing and scheduling problem in soft drink industries. *Pesquisa Operacional*, 27 (1), 2007, 155-186.
- [25] Toso, E. A. V., Morabito, R. and Clark, A. R. Lot sizing and sequencing optimization at an animal-feed plant. *Computers & Industrial Engineering*, 57, 2009, 813–821.