Ant algorithms for a time and space constrained assembly line balancing problem

Joaquín Bautista *, Jordi Pereira

Nissan Chair ETSEIB Universitat Politècnica de Catalunya, 7th Floor, Avda Diagonal 647, 08028 Barcelona, Spain

Available online 2 February 2006

Abstract

The present article focuses on the application of a procedure based on ant colonies to solve an assembly line balancing problem. After an introduction to assembly line problems, the problem under study is presented: the Time and Space constrained Assembly Line Balancing Problem (TSALBP); and a basic model of one of its variants is put forward for study. Subsequently, an ant algorithm is presented that incorporates some ideas that have offered good results with simple balancing problems. Finally, the validity of the proposed algorithms is tested by means of a computational experience with reference instances, and the conclusions of the study are presented.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Artificial intelligence; Manufacturing; Metaheuristics; Production

1. Introduction

An assembly line is made up of a number of workstations, \( m \), arranged in series and in parallel, through which the work in progress on a product flows. The stations are linked together by a transport system whose mission is to supply materials to the main flow and to move the production items from one station to the next. Production items may be of one type (single-model) or of several types (mixed-model).

The manufacturing of a production item is divided up into a set \( V \) of \( n \) tasks; each station \( k \) \((1 \leq k \leq m)\) is assigned a subset of tasks \( S_k \) \((S_k \subseteq V)\) called the workload of station \( k \); a task \( j \) may only be assigned to one station. Each task \( j \) requires an operation time \( t_j > 0 \) for its execution that is determined as a function of the manufacturing technologies and the employed resources. In mixed-model assembly lines, it is usual to

* Corresponding author. Tel.: +34 93 401 17 03; fax: +34 93 401 60 54.
E-mail addresses: joaquin.bautista@upc.edu (J. Bautista), jorge.pereira@upc.edu (J. Pereira).
calculate operation times weighting the presence of each model in the mix. If the range of required operation times between different models is high, units must be sequenced accordingly.

Additionally, the technology and nature of the product itself mean that each task \( j \) has a set of immediately predecessor tasks, \( P_j \), which must be accomplished before commencing task \( j \). These constraints are normally represented by means of an acyclic precedence graph whose vertices represent the tasks and in which each directed arc \((i,j)\) indicates that task \( i \) must be finished before commencing task \( j \) on the line; thus, if \( i \in S_h \) and \( j \in S_k \), then \( h \leq k \) must be fulfilled.

Each station \( k \) presents a station workload time \( t(S_k) \) that is equal to the sum of the durations of the tasks assigned to station \( k \). Once permanent manufacturing conditions have been achieved, the production items flow along the line at a constant rate, and each station \( k \) has a time \( c_k \), called the cycle time, to carry out its assigned tasks. On automotive trim or body lines, the items normally remain in spaces allotted to each station, where the corresponding tasks are executed during a time equal to the cycle. The items are then transferred to the next station in an insignificant period of time, thus initiating a new cycle.

The cycle time \( c \) determines the production rate \( r \) of the line \( (r = 1/c) \), and cannot be less than the maximum station workload time: \( c \geq \max_k \{ t(S_k) \} \), nor must it be greater than the sum of the durations of the tasks of \( V: c \leq \sum_k t(S_k) = t_{\text{sum}} \). Each station \( k \) presents a idle time \( I_k = c - t(S_k) \). The sum of these partial times gives rise to the total idle time, \( I_{\text{sum}} = \sum_k I_k = m \cdot c - t_{\text{sum}} \), which is related to the inefficiency of the line.

In general, ALBP (Assembly Line Balancing Problems) focus on grouping together the tasks in the set \( V \) in workstations in an efficient and coherent way. In short, the goal is to achieve a grouping of tasks that minimizes the inefficiency of the line or its total downtime and that respects all the constraints imposed on the tasks and on the stations. ALBP belongs to the general class of sequencing problems (see Baker, 1974), and can be seen as Bin Packing Problems with additional side constraints, as precedence relationships between tasks. These precedence constraints establish an implicit order of bins, deriving in a sequence of operations.

A first family of problems, known as SALBP (Simple Assembly Line Balancing Problems) Baybars (1986), may be stated in the following way: given a set of \( n \) tasks with their attributes and a precedence graph, each task must be assigned to a single station in such a way that all the precedence constraints are satisfied and no station workload time, \( t(S_k) \), is greater than the cycle time \( c \).

The SALBP family presents four variants: SALBP-1: minimize the number of stations \( m \) given a fixed value of the cycle time \( c \); SALBP-2: minimize the cycle time \( c \) (maximize the production rate \( r \)) given a fixed number of stations \( m \); SALBP-E: simultaneously minimize \( c \) and \( m \) considering their relation with the total idle time or the inefficiency of the line; SALBP-F: given \( m \) and \( c \), determine the feasibility of the problem, and if it is feasible, find a solution.

When other considerations are added to those of the SALBP family, the problems are known in the literature by the name of GALBP (General Assembly Line Balancing Problems). This family includes those problems with additional constraints, such as for instance the consideration of parallel stations, Daganzo and Blumfield (1994) and Vilarinho and Simaria (2002), forced groupings of tasks, Deckro (1989) possible incompatibilities among tasks, Agnetis et al. (1995) and differences between workstations, Nicosia et al. (2002), among others. An up-to-date analysis of the bibliography and available state of the art procedures can be found in Scholl and Becker (2006) for SALBP family of problems, and Becker and Scholl (2006) for GALBP ones.

As regards problem-solving procedures, the literature includes a large variety of these. A first group of algorithms is made up of those known as “greedy” algorithms, which are based on priority rules or partial enumerative procedures; see Talbot et al. (1986) and more recently Fleszar and Hindi (2003). A second group is composed of enumerative procedures, basically under a branch and bound paradigm, Johnson (1988), Hoffmann (1992), Scholl and Klein (1999), Sprecher (2003), currently being the most effective. And a third group composed of applications of diverse metaheuristics (see Scholl and Voss (1996)). Almost all these studies focus on the solution of SALBP-1 o SALBP-2 problems; for this reason, specific procedures must be employed when addressing a problem that includes differences with respect to said models.
The present paper is organized as follows. Section 2 presents a problem detected in the automotive sector that consists in the consideration of additional constraints related to the space available around the lines; a mathematical formulation is proposed and its relations with SALBP are established. In Section 3, an ant algorithm based on Dorigo et al. (1996) AS heuristic is put forward to solve the problem. The algorithm incorporates elements from other metaheuristics that have afforded good results in balancing problems. Section 4 deals with a computational experience on a collection of reference instances adapted to the new problem. Finally, the main conclusions of the study are presented in Section 5.

2. The time and space constrained assembly line balancing problem

2.1. Balancing assembly lines under space constraints

In a production system made up of several branched manufacturing and/or assembly lines, the balancing of lines is a frequent problem. This is the case of the automobile industry, in which it is common to use the same line for variants of a component (e.g. chassis, body, seats, etc.), or the same painting line for the bodies of different vehicles with their variants, or the same body line for variants of the same vehicle. The need for balancing is present in different situations, even if the balancing is temporary in nature. However, some of the causes for balancing an assembly line affects the allocation of surfaces allotted to these, on both sides of the line as well as in the areas of aerial transport. If an installation already exists the limitation of the space allotted to materials and to manufacturing and assembly tools will also have to be taken into account. Some of the common causes to take space into account are:

- The length of the workstation is limited. When the unit moves along the line at a constant rate, using a conveyor belt for example, the worker (or workers) starts its work as near to the initial point of the workstation as possible, and must fulfill their tasks while following the product. When the unit does not move while in a workstation and they are transferred to the following workstation regularly, at each cycle time, the worker needs to take the tools and materials to be assembled in the unit. In both cases, there are constraints to the maximum allowable movement of the worker that directly limit the length of the workstation, and the available space.
- The required tools and components to be assembled should be distributed along the sides of the line. It is important to keep them as near to the workplace as possible, as well as to avoid double lines of material. In addition to these common constraints, in the automotive industry, some operations can only be executed on one side of the line, conditioning the physical space where tools and materials can be allocated. Finally, several containers are very large, to reduce the number of deliveries or due to specialities of the component, as front and back seats. If several tasks requiring large areas for their supplies are put together, the workstation would be unfeasible.
- In mixed model assembly lines space constraints are even more important. This is a usual in the automotive industry, where regularization of components consumption led to different sequences from those originated in batch scheduling. When solving the balancing related problem, it is usual to use weighted duration of tasks, but each workstation will require the components and tools associated to each variant of product to be assembled in the workstation, increasing the allotted space for the workstation.
- Another usual source of spatial constraints comes from the change of products. When a car model is substituted for a newer one, it is usual to keep the same production plant. The production plant is usually designed taking into account the original model, but when it is substituted by a larger model or a more component-rich model, the additional requirement of space in the production plant will originate spatial constraints.
This limitation of a spatial nature may be contemplated by associating with each task \( j \) a required area \( a_j \) (a function of the cycle time \( c \), of the production mix and of the supply rate of materials) and by associating to each station \( k \) an available area \( A_k \) (a function of \( S_k \) and of the layout) that, for reasons of homogenization and simplification, we shall assume to be identical for all stations and equal to \( A \): \( A = A_k \) (\( 1 \leq k \leq m \)). Each station \( k \) requires a station area \( a(S_k) \) that is equal to the sum of areas required by the tasks assigned to station \( k \).

This leads us to a family of problems that we call TSALBP: Time and Space constrained Assembly Line Balancing Problems, which may be stated in the following way: given a set of \( n \) tasks with their temporal \( t_j \) and spatial \( a_j \) attributes (\( 1 \leq j \leq n \)) and a precedence graph, each task must be assigned to a single station such that: (1) all the precedence constraints are satisfied, (2) no station workload time, \( t(S_k) \), is greater than the cycle time, \( c \), and (3) no area required by the station, \( a(S_k) \), is greater than the available area per station, \( A \).

TSALBP presents diverse variants depending on the elements \( m \), \( c \) and \( A \). Here, we propose eight variants for TSALBP (see Table 1), depending on the elements \( m \), \( c \) and \( A \), which may act as fixed values or as optimization variables.

For example, TSALBP-1 (equivalent to SALBP-1) consists in minimizing the number of stations \( m \) given fixed values of the cycle time \( c \) and of the available area per station \( A \), whereas TSALBP-F is a feasibility problem given fixed values of \( m \), \( c \) and \( A \).

### 2.2. An example

Fig. 1 shows a precedence graph of the BOWMAN8 instance for the Simple Assembly Line Balancing Problem. The instance can be found in [http://www.assembly-line-balancing.de](http://www.assembly-line-balancing.de). The instance has eight tasks.

<table>
<thead>
<tr>
<th>Name</th>
<th>( m )</th>
<th>( c )</th>
<th>( A )</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSALBP-F</td>
<td>Given</td>
<td>Given</td>
<td>Given</td>
<td>F</td>
</tr>
<tr>
<td>TSALBP-1</td>
<td>Minimize</td>
<td>Given</td>
<td>Given</td>
<td>OP</td>
</tr>
<tr>
<td>TSALBP-2</td>
<td>Given</td>
<td>Minimize</td>
<td>Given</td>
<td>OP</td>
</tr>
<tr>
<td>TSALBP-3</td>
<td>Given</td>
<td>Given</td>
<td>Minimize</td>
<td>OP</td>
</tr>
<tr>
<td>TSALBP-1/2</td>
<td>Minimize</td>
<td>Minimize</td>
<td>Given</td>
<td>MOP</td>
</tr>
<tr>
<td>TSALBP-1/3</td>
<td>Minimize</td>
<td>Given</td>
<td>Minimize</td>
<td>MOP</td>
</tr>
<tr>
<td>TSALBP-2/3</td>
<td>Given</td>
<td>Minimize</td>
<td>Minimize</td>
<td>MOP</td>
</tr>
<tr>
<td>TSALBP-1/2/3</td>
<td>Minimize</td>
<td>Minimize</td>
<td>Minimize</td>
<td>MOP</td>
</tr>
</tbody>
</table>

The suffixes 1, 2 and 3 refer to the minimization of \( m \), \( c \) and \( A \), respectively. If the problem is one of feasibility, this is indicated by F in the column “Type”; if it is a mono-objective optimization problem, this is indicated by OP; if it is a multi-objective optimization problem, this is indicated by MOP. When the problem is MOP, the suffixes 1, 2 and 3 are concatenated with “/” to name the problem.

![Fig. 1. Precedence graph of the example. On the top of each task, vertex, processing time and required space are indicated.](image-url)
with processing times between 3 and 17 units. In addition to the processing times each task has been given a space requirement to show the TSALBP case. Both values appear in the top of each vertex, being the first value the required processing time, as in the original instance, and the area requirement, reversing the list of processing times.

Given a cycle time of 20, an optimal solution to the SALBP-1 instance requiring five stations is the following grouping of tasks: \{1\}, \{2\}, \{3, 5\}, \{4, 6, 8\}, \{7\}. Each station does not require more processing time than the cycle time and precedence relationships are fulfilled between tasks of different stations. If an additional space constraint is imposed, 20 units of space for each station, no solution exists with five stations. The previous solution does not fulfill the area constraint for station four, composed by tasks \{4, 6, 8\} requiring 28 units of space. As it can be seen, the combination of both constraints make no possible grouping requiring five stations, being the optimal solution the task grouping \{1\}, \{2\}, \{3, 5\}, \{4, 6\}, \{7\}, \{8\} requiring six stations.

2.3. A mathematical formulation for time and space constrained assembly line balancing problems

To formally describe a TSALBP model, we shall employ the following additional notation:

\begin{itemize}
  \item \(E_j, L_j\) The earliest and lastest station to which task \(j\) may be assigned.
  \item UB The upper bound of the number of stations.
  \item \(x_{jk}\) A decision variable equal to 1 if task \(j\) is assigned to station \(k\) (0 otherwise).
\end{itemize}

Under these conditions, expressions (1)–(6) establish the TSALBP-F model.

\begin{align*}
\sum_{k = E_j}^{L_j} x_{jk} &= 1 \quad 1 \leq j \leq n, \quad (1) \\
\sum_{k = 1}^{\text{UB}} \max\{x_{jk}\} &\leq m, \quad (2) \\
\sum_{j = 1}^{n} t_j x_{jk} &\leq c \quad 1 \leq k \leq \text{UB}, \quad (3) \\
\sum_{j = 1}^{n} a_j x_{jk} &\leq A \quad 1 \leq k \leq \text{UB}, \quad (4) \\
\sum_{k = E_i}^{L_i} k x_{ik} &\leq \sum_{k = E_j}^{L_j} k x_{jk} \quad (1 \leq i, j \leq n) \land (i \in P_j), \quad (5) \\
x_{ik} &\in \{0, 1\} \quad (1 \leq i \leq n) \land (1 \leq k \leq \text{UB}). \quad (6)
\end{align*}

Equalities (1) ensure that each task is assigned to only one station. Inequalities (2)–(4) respectively ensure that the number of stations with a workload do not exceed the permitted number, that the workload time at each station does not exceed the cycle time and that the required area at each station does not exceed the available area. Inequalities (5) ensure the fulfillment of the precedence relationships between tasks. Finally, constraints (6) define the decision variables as binary variables.

In TSALBP-F, the elements \(m, c\) and \(A\) act as parameters, whereas in the rest of the problems, one or more of these elements act as the objective function.
Expressions (7)–(9) correspond to the objective function of the TSALBP-1, TSALBP-2 and TSALBP-3, respectively.

\[
\text{minimize } Z_1(x) = m = \sum_{k=1}^{UB} \max_{1 \leq j \leq n} \{x_{jk}\},
\]

\[
\text{minimize } Z_2(x) = c = \max_{1 \leq k \leq UB} \left\{ \sum_{j=1}^{n} t_j x_{jk} \right\},
\]

\[
\text{minimize } Z_3(x) = A = \max_{1 \leq k \leq UB} \left\{ \sum_{j=1}^{n} a_j x_{jk} \right\}.
\]

The objectives of the MOP problems (TSALBP-1/2, TSALBP-1/3, TSALBP-2/3 and TSALBP-1/2/3) may be easily formalized on the basis of the elemental objectives established in expressions (7)–(9).

3. An ant algorithm for solving the TSALBP

The behavior of many species of ants in obtaining food is based on the depositing and following of trail via a chemical substance called pheromone. Given a source of food, an ant colony tends to naturally find the shortest path between the source and its nest via two processes. Firstly, the ants deposit pheromone on the route; and secondly, they normally follow the path on which they find more previously deposited pheromone. If they find a shorter path, more ants circulate along this path and more pheromone will be deposited along it. The adaptation of these ideas to the field of optimization is known as ant algorithms, which iteratively apply the following three phases: (1) solution building by means of a randomized procedure, (2) local improvement of said solutions, and (3) pheromone depositing to report back information for building of new solutions.

This section focuses on an application of ant algorithms for assembly line balancing problems with time and space constraints at the workstations. The algorithm presented here fits within the Ant System (AS) paradigm (Dorigo et al., 1996). We next describe in detail the overall schema of the algorithm, and then go on to show the three phases in which an ant algorithm is divided.

3.1. General schema of the procedure

Although the generation of solutions for solving similar problems to SALBP-1 is simple, it is difficult to define an efficient procedure of applicable local improvement; see Bautista and Pereira (2002), for a specific case, or Scholl and Voss (1996) for a more general discussion of these problems. The difficulty is related to the objective function of SALBP-1, where improvement moves can only be obtained when a workstation is left without tasks, and thus there is no obvious measure to find ‘good moves’ (see Scholl, 1999). For this reason, a solution strategy is proposed that is similar to that used for tabu search procedures (Scholl and Voss, 1996), for SALBP-1 that is based on the iterative solution of more and more restrictive SALBP-2 instances.

The procedure starts off by building an initial solution for a TSALBP-1 instance employing the heuristic rule presented in Section 3.2; a valid solution is thus obtained for a number of stations equal to \(m\). Subsequently, an AS procedure is applied to solve the TSALBP-2 instance with \(m - 1\) stations. If a solution is obtained with a maximum workload time and maximum required space equal to or less than the given values for the original TSALBP-1 instance, the found solution is better than the known one, thus reducing the number of stations required. This process may be applied yet again with a lower number of stations until reaching a final condition. To avoid stagnation in unpromising areas of the solutions space, a diversification mechanism is added based on reinitiating the pheromone information when, after generating a number of
solutions equal to 50 times the number of tasks in the instance without improving the current best solution. In such a case the AS procedure is restarted with the same number of workstations.

Although this iterative schema enables the application of local improvement algorithms (see Section 3.3) that are better directed than in a schema based on the direct solution of TSALBP-1, the generation of initial solutions is more complex.

The proposed procedure iteratively solves TSALBP-1 instances with the procedure shown in Section 3.2 until reaching a solution with the given number of stations is found. Initially, an attempt is made to build a solution for the instance with a given time and space equal to the cycle time and the available space, respectively. In case the solution offered by the procedure requires a greater number of stations than the desired number, the cycle time and available space is incremented by 1% or one unit, whatever greater, and a new initial solution is built until a solution with the desired number of stations is found. Likewise, when a solution is obtained with the required number of stations, cycle and available space are reduced with respect to their value between 1% of the given values or one unit, whatever is greater. Solutions with the desired number of stations undergo local improvement (see Section 3.3) and pheromone is deposited to build future solutions (Section 3.4). Algorithm 1 shows the proposed algorithm.

Algorithm 1. Procedure to solve TSALBP-1

\[
\begin{align*}
m &\leftarrow \text{heuristic}(c, A) \quad \text{//see Section 3.2} \\
\text{INIT:} & \\
\text{iteration} &\leftarrow 0 \\
c' &\leftarrow c \\
A' &\leftarrow A \\
m' &\leftarrow m - 1 \\
\text{RESTART:} & \\
\text{initialize_trail()} \quad \text{//see Section 3.4} \\
\text{do} & \\
m_c &\leftarrow \text{ants}(c', A') \quad \text{//see Section 3.2} \\
\text{IF}(m_c = m') \text{ then} & \\
(c', A') &\leftarrow \text{localsearch()} \quad \text{//see Section 3.3} \\
\text{IF } c' \leq c \text{ AND } A' \leq A & \\
m &\leftarrow m' \\
\text{Goto INIT} \\
\text{ENDIF} & \\
\text{iteration} &\leftarrow \text{iteration} + 1 \\
\text{IF iteration} = 50 \times n \text{ GOTO RESTART} & \\
\text{Update_trail}(c', A') \quad \text{//see Section 3.4} & \\
c' &\leftarrow \text{MIN}(|c' \times 0.99|, c' - 1) \\
A' &\leftarrow \text{MIN}(|A' \times 0.99|, A' - 1) & \\
\text{ELSE} & \\
c' &\leftarrow \text{MAX}(|c' \times 1.01|, c' + 1) \\
A' &\leftarrow \text{MAX}(|A' \times 1.01|, A' + 1) & \\
\text{ENDIF} & \\
\text{WHILE(\text{NOT}(\text{end_condition}))} & \\
\end{align*}
\]

We next show the way of generating solutions based on the schema for SALBP-1 problems from the literature. We then go on to show how local improvement is carried out, according to the SALBP-2 schema, and the techniques for managing the trail used by the building procedure.
3.2. Solution generation

The majority of building procedures for the solving of balancing problems are based on the application of priority rules or constrained enumerative procedures. An in-depth study of the available procedures can be found in Scholl (1999).

Procedures based on priority rules are based on the building of feasible solutions for SALBP-1 instances. By means of a series of rules, each task is assigned a priority value that depends on its process time and its precedence relations (see Hackman et al., 1989). The given priorities are used in the building process to choose the tasks to assign to the stations.

The literature includes two frameworks for solution building: one that is station-oriented and another that is task-oriented. The former is used in this study, as diverse computational experiments for SALBP-1 (see Scholl and Voss, 1996), indicate that the station-oriented procedure generally provides better results.

The procedure is initiated by the opening of a first station \( k = 1 \). Tasks are then successively assigned to this station until more tasks cannot be assigned, in which case, said station is closed and a new station is opened. In each iteration, the candidate task with the greatest priority is assigned to the current station; a task is a candidate when its preceding tasks have been assigned and it requires less time and space that those available in the station under construction. When no more tasks may be assigned to the open station, this is closed and the following station \( k + 1 \) is opened. The procedure finalizes when there are no more tasks left to assign.

For the building procedures put forward in this paper, a mixed rule is employed that gives joint priority to the duration of a task and the number of tasks succeeding it. The required space is also taken into consideration in this adaptation, and the values of the three elements that make up the rule (space, time and successors) is normalized so as to achieve a stable range of values that may be used for any instance of the problem. This enables the ant algorithm to be used with visibility values similar for all the instances solved.

In general, the heuristic weight of a task, \( j \), \( \eta_j \), is equal to

\[
\eta_j = \frac{a_j}{A} + \frac{t_j}{c} + \frac{F_j^*}{\max_{i \in F_j^*}|F_i^*|},
\]

where \( F_j^* \) is the set of all successors of task \( j \).

In the case of using a probabilistic building schema, as is characteristic of ant algorithms, it is necessary to modify the station-oriented procedure described above. The probabilistic selection criterion must depend on the heuristic information of the task, \( \eta_j \), and on the information provided by the pheromone deposited by previous solutions, \( \tau_{kj} \). The pheromone may be used in two ways, either directly, relating a task, \( j \), with the station under construction, \( k \), or accumulatively, relating a task, \( j \), with the station under construction, \( k \), and with those already closed \( (h < k) \), as in Merkle et al. (2000).

In the case of direct evaluation, the likelihood of selecting candidate task \( j \) belonging to the set \( D_k \) of candidate tasks to be assigned to station \( k \), is determined in the following way:

\[
p_{kj} = \frac{[\tau_{kj}]^\alpha [\eta_j]^\beta}{\sum_{i \in D_k} [\tau_{ki}]^\alpha [\eta_i]^\beta},
\]

where \( \tau_{kj} \) and \( \tau_{ki} \) correspond to the pheromone deposited by previous solutions and \( \eta_j \), \( \eta_i \) to the heuristic weight of each task.

In the case of accumulative evaluation, the likelihood of selection is determined as follows:

\[
p_{kj} = \frac{\left(\sum_{h=1}^{k}[\tau_{kh}]\right)^\alpha [\eta_j]^\beta}{\sum_{i \in D_k} \left(\sum_{h=1}^{k}[\tau_{hi}]\right)^\alpha [\eta_i]^\beta},
\]

where \( h \) refers to the stations that have already been closed or to the open station in the current solution.
In both cases, \( x \) and \( \beta \) are constants that respectively define the influence of the heuristic component and the trail information on the decision of the ants.

### 3.3. Local improvement procedure

A local improvement is applied to the solutions offered by the building procedure that transforms one feasible solution into others.

The improvement procedures for balancing problems are based on exchanges and movements (see Rachamadugu and Talbot, 1991). To explain these procedures, it is necessary to define for each task \( j \) the first, \( ES_j \), and last, \( LP_j \), station of the current iteration between which a task may be assigned in consonance with the assignment of its immediate predecessors and successors.

In general, a movement \((j, k_1, k_2)\) describes the movement of the task \( j \) from station \( k_1 \) to station \( k_2 \), where \( k_1 \neq k_2 \) and \( k_2 \in [ES_j, LP_j] \), whereas an exchange \((j_1, k_1, j_2, k_2)\) simultaneously moves task \( j_1 \) from station \( k_1 \) to \( k_2 \) and task \( j_2 \) from station \( k_2 \) to \( k_1 \). Obviously, tasks \( j_1 \) and \( j_2 \) are not related by precedence and stations \( k_1 \) and \( k_2 \) are distinct. The described exchange is feasible if the movements \((j_1, k_1, k_2)\), \((j_2, k_2, k_1)\) are feasible.

Furthermore, the set of candidate tasks that may improve the current solution is made up of those tasks that form part of the stations with the greatest occupation as regards cycle time or available space, and these are therefore the only tasks that must be taken into account in the local improvement procedure. In the implementation carried out, one of the critical stations with respect to duration is identified and another with respect to space, and all the possible transformations are tested, both of movements as well as exchanges. Those transformations that improve the value provided by said station in the solution without worsening the overall solution are carried out immediately. The improvement algorithm is detained if the neighborhood of a solution does not afford improvements or when a solution is obtained that is compatible with the space and cycle time given to the instance.

### 3.4. Pheromone maintenance

Initially, the pheromone matrix contains a constant and equal value for all possibilities; a value of 0.5 being established in the present case.

Whenever a solution is built with the desired number of stations, and after applying the local improvement procedure, evaporation of a fixed proportion, \( \rho \), of the pheromone deposited on the trail matrix begins; that is:

\[
\tau_{kj} = (1 - \rho) \cdot \tau_{kj}. \tag{13}
\]

Subsequently, pheromone is deposited according to the quality of the obtained solution and equal to:

\[
\tau_{kj} = \rho \cdot \frac{c^* + A^*}{c^* + A^*} + \tau_{kj}, \tag{14}
\]

where \( c^* \) and \( A^* \), stand for the cycle time and required area of the solution depositing pheromone and \( c^* \) and \( A^* \) are the imposed cycle time and available area for the instance.

### 4. Computational experience

In order to compare the result of the proposed procedures, different computational experiences were carried out with two collection of instances from the literature (see http://www.assembly-line-balancing.de).
made up of 269 and 302 instances for SALBP-1 and SALBP-2 respectively. A final experiment is done using a real life instance provided by the Barcelona Plant of Nissan, in Spain.

The first computational experience, used to check the validity of the proposed Ant System, is based on the solution of the SALBP-2 problem with the inner ant system algorithm. To do so, the number of stations is fixed to the parameter of the instance and the best cycle time obtained is recorded. As the algorithm is prepared to improve cycle and space altogether, available space at the stations and the cycle time, $A = c$, is made equal in all the instances, as is the required space and the duration of each task, $a_j = t_j$. In this case, the solution coincides with the equivalent SALBP-2 instance.

The best set of control parameters was previously determined, as well as the best way of reading the pheromone (accumulatively or directly), obtaining as the best combination among those tested values of $\alpha = 5$, $\beta = 1$ and $\rho = 0.1$, employing direct reading.

It is interesting to note the reversibility of Assembly Line Balancing Problems. That is, if precedence relationships are reversed, a solution to the instance is also a valid solution to the original instance. As indicated in Hoffmann (1992) and Scholl (1999), a solution procedure may provide different solution for the direct and reverse instance, so it is recommendable to solve both instances or to devise a procedure taking this property into account. Our approach solves direct and reverse instance separately reporting the solutions found by each separately as well as the best solution found.

Table 2 shows the results obtained with a maximum of 30 seconds computing time, using a 1.8 GHz computer running LINUX. The algorithm was implemented in C and compiled with GCC version 3.2. The procedure was stopped when known optimal solutions were found or a maximum computation time of 30 seconds was reached. Each instance in the collection is solved using the proposed algorithm and compared to a tabu search due to Scholl and Voss (1996).

The proposed algorithm finds better solutions than the best known heuristic for the problem. This lead us to a second computational experience, used to check the validity of the proposed search procedure, based on the solution of SALBP-1 problem with the Algorithm 1. To do so, the available space at the stations and the cycle time is made equal in all the instances, as is the required space and the duration of each task. In this case, a solution to the TSALBP-1 instance coincides with the solution to the equivalent SALBP-1 instance.

Table 3 shows the results obtained by the proposed ant algorithms compared with the exact procedure SALOME two tabu search implementations (see Scholl, 1999), with a calculation time equal to 500 seconds on a PC 80486 DX2-66 and a previous ant algorithm proposed by the authors, denoted as ANTS-O designed to solve SALBP-1 with incompatibilities between tasks (see Bautista and Pereira, 2002).

The main differences between ANTS-O and the new approach are based on the solution generation, local search phase. ANTS-O uses 13 different priority rules to build solutions for SALBP-1 instances. As different priority rules are used, the pheromone plays an extra role as solutions found with one priority rule can

Table 2
SALBP-2 results

<table>
<thead>
<tr>
<th></th>
<th>Tabu search</th>
<th>ANTS-D</th>
<th>ANTS-R</th>
<th>ANTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>#opt</td>
<td>165</td>
<td>180</td>
<td>176</td>
<td>199</td>
</tr>
<tr>
<td>Av. dev. (%)</td>
<td>0.66</td>
<td>1.08</td>
<td>1.26</td>
<td>0.75</td>
</tr>
<tr>
<td>Max. dev. (%)</td>
<td>7.94</td>
<td>8.69</td>
<td>10</td>
<td>7.24</td>
</tr>
</tbody>
</table>

Results are reported for the best available results for tabu search (see Scholl and Voss, 1996), and the proposed algorithm applied to the direct instances, ANTS-D, the reverse instances, ANTS-R, and to both instances, ANTS. Number of optimal solutions (#opt), average deviation (Av. dev. (%)) and maximum deviation (Max. dev. (%)) to optimal solution, or best bound available, are reported for each algorithm. Average and maximum deviation has been calculated as the mean for each instance of the difference between the solution found and best known solution divided by the best known solution.
communicate information for building solutions using other priority rules, and thus combining rules for solution generation. A SALBP-1 local search was also proposed by it showed to be inefficient.

Each instance in the collection is solved considering its original precedence graph (ANT-D) and its inverted precedence graph (ANT-R), imposing two stopping conditions on the ant algorithm: a maximum computation time of 30 seconds or the obtainment of the optimum solution. Obviously, the best solution among the solutions of each instance is also taken into account, considering its precedence graph in the direct or inverse sense (ANTS), registering in this case a computation time equal to the sum of both solutions. In the case of ANTS-O each instance is solved considering that no incompatibilities exist and imposing as a stopping condition a maximum computation time of 300 seconds using a 600 MHz. computer running Windows98.

As can be seen, the ant algorithm obtains similar solutions to the best implementation presented by an exact algorithm for SALBP-1, and obtains clearly superior results to the best specific metaheuristic for the problem, including the EurTabu heuristic, which incorporates an implicit enumeration step in the building of the initial solution, and outperforms the previous ant algorithm. This leads us to think on the importance of the selected procedure to obtain a state of the art procedure to SALBP-1. It is important to note that running times between the ant algorithm and the other procedures is not comparable due to the difference between the computers used for the computational experience. Anyhow the viability of an ant algorithm approach to solve Assembly Line Balancing Problems and the quality of the solutions provided by the proposed procedure is demonstrated, even if the procedure has been designed to solve a more general class of problems than SALBP.

A second computational experience with SALBP-1 instances is conducted to verify which elements of the procedure are the most successful to solve the problem. Three different variations of the algorithm have been tested. The first one is restricted to the generation procedure without a local search phase and diversification mechanism, a second one does not count with the diversification mechanism and the final one is the final version used in the previous computational experience. We stop the search after 30 seconds and use the same control parameters as before for the ant algorithm. Table 4 shows the number of optimal solutions found by each proposed procedure.

As Table 4 shows, all the proposed elements are important for finding good solutions. The diversification strategy seems to play a key role to the found results as it allows to search on a broad number of area of the solution space rather than concentrating on a single neighbourhood. As several solutions with the same number of workstations exist for an instance, the diversification scheme allows to find other search areas where optimal or near optimal solutions are easier to reach using the local search procedure. If we compare the results with the number of optimal solutions found by the previous Ants Approach from the authors, Bautista and Pereira (2002), the previous results are outperformed by any combination of elements.

<table>
<thead>
<tr>
<th></th>
<th>SALOME</th>
<th>PrioTabu</th>
<th>EurTabu</th>
<th>ANTS-O</th>
<th>ANTS-D</th>
<th>ANTS-R</th>
<th>ANTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>#opt</td>
<td>227</td>
<td>200</td>
<td>214</td>
<td>182</td>
<td>218</td>
<td>214</td>
<td>227</td>
</tr>
<tr>
<td>Av. dev. (%)</td>
<td>0.46</td>
<td>0.86</td>
<td>0.63</td>
<td>0.4</td>
<td>0.7</td>
<td>0.8</td>
<td>0.6</td>
</tr>
<tr>
<td>Max. dev. (%)</td>
<td>7.69</td>
<td>7.69</td>
<td>7.69</td>
<td>N/A</td>
<td>14.28</td>
<td>14.28</td>
<td>14.28</td>
</tr>
<tr>
<td>Av. CPU</td>
<td>98.6</td>
<td>101.8</td>
<td>62.6</td>
<td>300</td>
<td>6.53</td>
<td>7.31</td>
<td>13.84</td>
</tr>
</tbody>
</table>

Average and maximum deviation has been calculated as the mean for each instance of the difference between the solution found and best known solution divided by the best known solution. Results are reported for the Branch and Bound procedure SALOME (see Scholl, 1999), two Tabu Search implementations, PrioTabu and EurTabu (see Scholl and Voss, 1996), a previous ant algorithm approach by the authors (Bautista and Pereira, 2002), and the proposed algorithm applied to the direct instances, ANTS-D, the reverse instances, ANTS-R, and to both instances, ANTS.
The final computational experience focuses on the solution of the TSALBP-1 problem. In order to facilitate comparison of the proposed algorithm with other future algorithms, the same collection of instances was employed, while adding: the required space for each task $a_i = \frac{t_i}{C_0} + 1 (1 \leq i \leq n)$ and an available space in each station equal to the cycle time.

The same control parameters were used in the experiment as those employed for the SALBP-1 case, with a maximum computation time equal to 2 minutes per instance, comparing the obtained values with the best known solutions for the SALBP-1 situation (Table 5).

From the analysis of the above results, it can be seen that TSALBP instances require a greater number of stations than SALBP instances, although some instances require exactly the same number of stations. As the proposed algorithm is capable of obtaining optimal or near optimal solutions to SALBP-1 instances, it is feasible to compare the differences between the solutions for TSALBP-1 and SALBP-1 in number of required workstations. This leads us to think that the increase in the number of required workstations appreciated will make available bounds for SALBP-1 instances inefficient for TSALBP-1 instances.

As the best available procedures to solve SALBP are Branch and Bound implementations, new techniques are required to solve TSALBP problems. This results is in accordance with other investigations (see Becker and Scholl, 2006), for example, where the authors state that highly developed enumeration and bounding schemes as well as metaheuristics will be required to solve non SALBP assembly line balancing problems.

Finally, the algorithm was tested on data provided by the Nissan Barcelona Plant. Appendix A shows an instance originated in the final assembly phase of the Nissan Pathfinder engines (a recently launched model). The table in Appendix A shows the task number ($n$), internal identifier from NISSAN (Id.), duration of the task ($t$) in seconds, required area ($a$) in meters, and precedence constraints of each task. Some changes have been made to the data:

- The original line corresponds to a mixed-model assembly line. Following the procedure in use in NISSAN, the duration of tasks has been modified taking into account the expected production mix of the variants to assemble. We remark that the production mix does not alter the area required for each task.
- The requirements of space originated by tools and machinery required for assembly have been omitted. Due to the similitude of the tasks and low cost of the machinery in use, each workstation is considered to contain all tools and thus the required space for them can be subtracted from the total available space for a workstation.

Table 4
Number of optimal solutions (#opt) are reported for each algorithm

<table>
<thead>
<tr>
<th>#opt</th>
<th>Heuristic</th>
<th>NLs-NDiv</th>
<th>Ls-NDiv</th>
<th>ANTS-D</th>
<th>ANTS</th>
<th>ANTS-O</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>153</td>
<td>185</td>
<td>191</td>
<td>218</td>
<td>227</td>
<td>185</td>
</tr>
</tbody>
</table>

The number of optimal solutions found by the heuristic priority rule (Heuristic), the proposed procedure with no local search or diversification technique (NLs-NDiv), with local search and without diversification procedure (Ls-NDiv) and the complete procedure is reported (ANTS-D). As a measure of reference the solution found solving both the direct and reverse instance is also reported (ANTS) and a previous ant algorithm (ANT-O) approach by the authors, Bautista and Pereira (2002).

Table 5
Number of optimal solutions (#opt), mean average deviation (Av. dev. (%)), and mean and maximum increment of stations, Av. increase and Max. increase respectively, with respect to SALBP-1 instances for the proposed algorithm

<table>
<thead>
<tr>
<th></th>
<th>ANTS-D</th>
<th>ANTS-R</th>
<th>ANTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>#opt</td>
<td>49</td>
<td>45</td>
<td>52</td>
</tr>
<tr>
<td>Av. dev. (%)</td>
<td>11.47</td>
<td>12.02</td>
<td>11.06</td>
</tr>
<tr>
<td>Av. increase</td>
<td>2.44</td>
<td>2.62</td>
<td>2.39</td>
</tr>
<tr>
<td>Max. increase</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
</tbody>
</table>
The cycle time of the line is 180 seconds while the available area per workstation is 4 m.

The duration, required area and precedence constraints of tasks have been slightly altered due to confidentiality issues.

The solution obtained is also reported in Appendix A for further comparisons, as well as the solution obtained by neglecting space constraints. As it can be seen, the number of required workstations differ from one solution to the other by four workstations (the SALBP-1 instance has 17 workstations, while the equivalent TSALBP-1 instance requires 21 workstations).

5. Conclusions

This paper presents a new family of assembly line balancing problems, called Time and Space constrained Assembly Line Balancing Problems, that arise in the automobile industry, among others, due to alterations in demand. A model is formulated for one of its possible variants. An ant algorithm is then proposed for solving the problem. The algorithm is tested comparing the solutions offered with those found in the literature for the SALBP-1 and SALBP-2 case, obtaining the best results in the literature from heuristics. The results offered for the TSALBP-1 case are presented, showing the greater computational difficulty of the TSALBP case. Finally, a real life instance is put forward and the solution obtained by the proposed procedure is shown.

Acknowledgements

We acknowledge the Spanish Operation of Nissan as well as the UPC Nissan Chair for partially funding this research work as well as providing real data. This research paper has also been partially funded by DPI2004-03475 Grant from the Spanish government. We are also grateful to two anonymous referees and the editor of this special issue for providing us with several suggestions which have improved the quality of the paper.

Appendix A

<table>
<thead>
<tr>
<th>j</th>
<th>Id.</th>
<th>t</th>
<th>a</th>
<th>P</th>
<th>SALBP-1</th>
<th>TSALBP-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50100</td>
<td>60</td>
<td>3</td>
<td>3,31</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>50110</td>
<td>75</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>50120</td>
<td>20</td>
<td>0,5</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>50500</td>
<td>60</td>
<td>1</td>
<td>3,5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>50501</td>
<td>20</td>
<td>0,5</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>50600</td>
<td>60</td>
<td>1,5</td>
<td>4,5</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>50800</td>
<td>45</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>50900</td>
<td>10</td>
<td>0,5</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>51000</td>
<td>20</td>
<td>0,5</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>51200</td>
<td>30</td>
<td>0,5</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>51400</td>
<td>15</td>
<td>0,5</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>51401</td>
<td>15</td>
<td>0,5</td>
<td>11</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>51600</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>51800</td>
<td>10</td>
<td>0,5</td>
<td>3,13</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>52000</td>
<td>8</td>
<td>1</td>
<td>9,10,11,13,14</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
### Appendix A (continued)

<table>
<thead>
<tr>
<th>$j$</th>
<th>Id.</th>
<th>$t$</th>
<th>$a$</th>
<th>$P$</th>
<th>SALBP-1</th>
<th>TSALBP-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>52010</td>
<td>8</td>
<td>0.5</td>
<td>9,10,11,13,14</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>17</td>
<td>52200</td>
<td>80</td>
<td>1</td>
<td>9,10,11,13,14</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>18</td>
<td>52400</td>
<td>40</td>
<td>0.5</td>
<td>9,10,11,13,14</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>19</td>
<td>52600</td>
<td>5</td>
<td>0.5</td>
<td>9,10,11,13,14</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>52610</td>
<td>5</td>
<td>0.5</td>
<td>9,10,11,13,14</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>21</td>
<td>52650</td>
<td>5</td>
<td>0.5</td>
<td>9,10,11,13,14</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>22</td>
<td>52700</td>
<td>7</td>
<td>0.5</td>
<td>26,27</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>23</td>
<td>52710</td>
<td>7</td>
<td>0.5</td>
<td>26,27</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>24</td>
<td>52720</td>
<td>30</td>
<td>0.5</td>
<td>26,27</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>25</td>
<td>52730</td>
<td>30</td>
<td>0.5</td>
<td>26,27</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>26</td>
<td>52750</td>
<td>5</td>
<td>0.5</td>
<td>15,16,17,18,19,20,21</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>27</td>
<td>52760</td>
<td>5</td>
<td>0.5</td>
<td>15,16,17,18,19,20,21</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>28</td>
<td>52800</td>
<td>30</td>
<td>1</td>
<td>22,23,24,25</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>29</td>
<td>52820</td>
<td>10</td>
<td>0.5</td>
<td>28</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>30</td>
<td>52900</td>
<td>15</td>
<td>1</td>
<td>29</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>31</td>
<td>52901</td>
<td>10</td>
<td>0</td>
<td>6,7,8,30</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>32</td>
<td>53050</td>
<td>15</td>
<td>0.5</td>
<td>31</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>33</td>
<td>53100</td>
<td>30</td>
<td>1</td>
<td>32</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>34</td>
<td>53200</td>
<td>10</td>
<td>0.5</td>
<td>32</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>35</td>
<td>53300</td>
<td>5</td>
<td>0.5</td>
<td>36</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>36</td>
<td>53301</td>
<td>25</td>
<td>1</td>
<td>32</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>37</td>
<td>53400</td>
<td>15</td>
<td>0</td>
<td>32,35</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>38</td>
<td>53600</td>
<td>5</td>
<td>0.5</td>
<td>33,34,36,37</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>39</td>
<td>53630</td>
<td>5</td>
<td>0.5</td>
<td>33,34,36,37</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>40</td>
<td>53650</td>
<td>5</td>
<td>0.5</td>
<td>33,34,36,37</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>41</td>
<td>54000</td>
<td>60</td>
<td>0.5</td>
<td>38,39,40</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>42</td>
<td>54100</td>
<td>15</td>
<td>1.5</td>
<td>38,39,40</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>43</td>
<td>54120</td>
<td>15</td>
<td>1.5</td>
<td>38,39,40</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>44</td>
<td>54200</td>
<td>25</td>
<td>0.5</td>
<td>41,42,43</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>45</td>
<td>54210</td>
<td>25</td>
<td>0.5</td>
<td>41,42,43</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>46</td>
<td>54230</td>
<td>5</td>
<td>0.5</td>
<td>44,45</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>47</td>
<td>54240</td>
<td>35</td>
<td>0.5</td>
<td>46</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>48</td>
<td>54250</td>
<td>35</td>
<td>0.5</td>
<td>46</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>49</td>
<td>54260</td>
<td>5</td>
<td>0.5</td>
<td>42,43</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>50</td>
<td>54270</td>
<td>15</td>
<td>0.5</td>
<td>47,48,49</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>51</td>
<td>54280</td>
<td>25</td>
<td>0</td>
<td>47,48,49</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>52</td>
<td>54290</td>
<td>30</td>
<td>0</td>
<td>47,48,49</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>53</td>
<td>54300</td>
<td>15</td>
<td>0</td>
<td>47,48,49</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>54</td>
<td>54310</td>
<td>15</td>
<td>0</td>
<td>47,48,49</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>55</td>
<td>54320</td>
<td>20</td>
<td>0</td>
<td>47,48,49</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>56</td>
<td>54330</td>
<td>10</td>
<td>0</td>
<td>47,48,49</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>57</td>
<td>54370</td>
<td>10</td>
<td>0.5</td>
<td>50,51,52,53,54,55,56</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>58</td>
<td>54500</td>
<td>20</td>
<td>0.5</td>
<td>57,59,60</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>59</td>
<td>54501</td>
<td>5</td>
<td>0</td>
<td>41</td>
<td>6</td>
<td>8</td>
</tr>
</tbody>
</table>

(continued on next page)
<table>
<thead>
<tr>
<th>j</th>
<th>Id.</th>
<th>t</th>
<th>α</th>
<th>P</th>
<th>SALBP-1</th>
<th>TSALBP-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>54520</td>
<td>20</td>
<td>0.5</td>
<td>42.43</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>61</td>
<td>54700</td>
<td>45</td>
<td>1</td>
<td>57.58</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>62</td>
<td>54720</td>
<td>30</td>
<td>0.5</td>
<td>61</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>63</td>
<td>54800</td>
<td>30</td>
<td>0.5</td>
<td>57</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>64</td>
<td>54820</td>
<td>10</td>
<td>0.5</td>
<td>57</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>65</td>
<td>55050</td>
<td>5</td>
<td></td>
<td>61,62,63,64</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>66</td>
<td>55200</td>
<td>10</td>
<td>0.5</td>
<td>61,62,63,64</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>67</td>
<td>55250</td>
<td>15</td>
<td>0.5</td>
<td>66</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>68</td>
<td>55300</td>
<td>60</td>
<td>1.5</td>
<td>65,67</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>69</td>
<td>55350</td>
<td>10</td>
<td>0.5</td>
<td>68</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>70</td>
<td>55400</td>
<td>30</td>
<td>1</td>
<td>67</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>71</td>
<td>55500</td>
<td>10</td>
<td>0.5</td>
<td>68</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>72</td>
<td>55540</td>
<td>10</td>
<td>0.5</td>
<td>68</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>73</td>
<td>55800</td>
<td>40</td>
<td>1.5</td>
<td>71,72</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>74</td>
<td>55900</td>
<td>25</td>
<td>0.5</td>
<td>68,69,70,73</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>75</td>
<td>56000</td>
<td>10</td>
<td>0.5</td>
<td>74</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>76</td>
<td>56020</td>
<td>10</td>
<td>1</td>
<td>74</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>77</td>
<td>56100</td>
<td>15</td>
<td>0.5</td>
<td>75</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>78</td>
<td>56200</td>
<td>15</td>
<td>0.5</td>
<td>79</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>79</td>
<td>56220</td>
<td>15</td>
<td>0.5</td>
<td>74</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>80</td>
<td>56300</td>
<td>10</td>
<td>0.5</td>
<td>76,77,78</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>81</td>
<td>56400</td>
<td>10</td>
<td>1</td>
<td>76,77,78</td>
<td>10</td>
<td>14</td>
</tr>
<tr>
<td>82</td>
<td>56401</td>
<td>10</td>
<td>0</td>
<td>80,81</td>
<td>10</td>
<td>14</td>
</tr>
<tr>
<td>83</td>
<td>56420</td>
<td>20</td>
<td>0.5</td>
<td>82</td>
<td>10</td>
<td>14</td>
</tr>
<tr>
<td>84</td>
<td>56430</td>
<td>10</td>
<td>0</td>
<td>83</td>
<td>10</td>
<td>14</td>
</tr>
<tr>
<td>85</td>
<td>56440</td>
<td>20</td>
<td>0.5</td>
<td>75,84</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>86</td>
<td>56500</td>
<td>25</td>
<td>0.5</td>
<td>82</td>
<td>10</td>
<td>14</td>
</tr>
<tr>
<td>87</td>
<td>56600</td>
<td>20</td>
<td>0.5</td>
<td>82</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>88</td>
<td>56700</td>
<td>15</td>
<td>0.25</td>
<td>84</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>89</td>
<td>56750</td>
<td>20</td>
<td>0.5</td>
<td>88</td>
<td>11</td>
<td>15</td>
</tr>
<tr>
<td>90</td>
<td>56760</td>
<td>30</td>
<td>0.5</td>
<td>88</td>
<td>11</td>
<td>15</td>
</tr>
<tr>
<td>91</td>
<td>56800</td>
<td>20</td>
<td>0.5</td>
<td>85,86,87,88</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>92</td>
<td>56880</td>
<td>25</td>
<td>0.5</td>
<td>89,90,91</td>
<td>11</td>
<td>15</td>
</tr>
<tr>
<td>93</td>
<td>56900</td>
<td>10</td>
<td>0.5</td>
<td>92</td>
<td>13</td>
<td>15</td>
</tr>
<tr>
<td>94</td>
<td>56920</td>
<td>5</td>
<td>0.5</td>
<td>89,90,91</td>
<td>13</td>
<td>15</td>
</tr>
<tr>
<td>95</td>
<td>56940</td>
<td>20</td>
<td>0.5</td>
<td>94</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>96</td>
<td>57000</td>
<td>10</td>
<td>0.5</td>
<td>93,95,99</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>97</td>
<td>57050</td>
<td>5</td>
<td>0.5</td>
<td>93,95,99</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>98</td>
<td>57100</td>
<td>80</td>
<td>0</td>
<td>92</td>
<td>12</td>
<td>16</td>
</tr>
<tr>
<td>99</td>
<td>57120</td>
<td>30</td>
<td>0</td>
<td>89,90,91</td>
<td>11</td>
<td>15</td>
</tr>
<tr>
<td>100</td>
<td>57150</td>
<td>10</td>
<td>0.5</td>
<td>98,99</td>
<td>12</td>
<td>16</td>
</tr>
<tr>
<td>101</td>
<td>57160</td>
<td>10</td>
<td>0.5</td>
<td>98,99</td>
<td>12</td>
<td>16</td>
</tr>
<tr>
<td>102</td>
<td>57200</td>
<td>20</td>
<td>0.5</td>
<td>100,101</td>
<td>12</td>
<td>16</td>
</tr>
<tr>
<td>103</td>
<td>57210</td>
<td>30</td>
<td>0.5</td>
<td>100,101</td>
<td>12</td>
<td>16</td>
</tr>
<tr>
<td>104</td>
<td>57250</td>
<td>5</td>
<td>0</td>
<td>102,103</td>
<td>12</td>
<td>17</td>
</tr>
</tbody>
</table>
### Appendix A (continued)

<table>
<thead>
<tr>
<th>j</th>
<th>Id.</th>
<th>t</th>
<th>a</th>
<th>P</th>
<th>SALBP-1</th>
<th>TSALBP-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>105</td>
<td>57300</td>
<td>30</td>
<td>0,5</td>
<td>106</td>
<td>13</td>
<td>17</td>
</tr>
<tr>
<td>106</td>
<td>57301</td>
<td>25</td>
<td>0,5</td>
<td>100,101</td>
<td>12</td>
<td>16</td>
</tr>
<tr>
<td>107</td>
<td>57400</td>
<td>5</td>
<td>0</td>
<td>100,101,104</td>
<td>15</td>
<td>17</td>
</tr>
<tr>
<td>108</td>
<td>57450</td>
<td>5</td>
<td>0</td>
<td>100,101,104</td>
<td>13</td>
<td>17</td>
</tr>
<tr>
<td>109</td>
<td>57500</td>
<td>5</td>
<td>0,5</td>
<td>108</td>
<td>13</td>
<td>17</td>
</tr>
<tr>
<td>110</td>
<td>57505</td>
<td>5</td>
<td>0</td>
<td>108</td>
<td>13</td>
<td>17</td>
</tr>
<tr>
<td>111</td>
<td>57510</td>
<td>10</td>
<td>0</td>
<td>109,110</td>
<td>13</td>
<td>17</td>
</tr>
<tr>
<td>112</td>
<td>57520</td>
<td>10</td>
<td>0</td>
<td>109,110</td>
<td>13</td>
<td>17</td>
</tr>
<tr>
<td>113</td>
<td>57530</td>
<td>15</td>
<td>0,5</td>
<td>108</td>
<td>13</td>
<td>17</td>
</tr>
<tr>
<td>114</td>
<td>57540</td>
<td>20</td>
<td>0</td>
<td>113</td>
<td>13</td>
<td>17</td>
</tr>
<tr>
<td>115</td>
<td>57550</td>
<td>20</td>
<td>0</td>
<td>113</td>
<td>13</td>
<td>17</td>
</tr>
<tr>
<td>116</td>
<td>57700</td>
<td>45</td>
<td>1</td>
<td>111,112,114,115</td>
<td>13</td>
<td>17</td>
</tr>
<tr>
<td>117</td>
<td>57900</td>
<td>20</td>
<td>0,5</td>
<td>118</td>
<td>14</td>
<td>18</td>
</tr>
<tr>
<td>118</td>
<td>57950</td>
<td>25</td>
<td>0</td>
<td>116</td>
<td>14</td>
<td>18</td>
</tr>
<tr>
<td>119</td>
<td>58000</td>
<td>25</td>
<td>0</td>
<td>116</td>
<td>14</td>
<td>18</td>
</tr>
<tr>
<td>120</td>
<td>58050</td>
<td>20</td>
<td>0,5</td>
<td>119</td>
<td>14</td>
<td>18</td>
</tr>
<tr>
<td>121</td>
<td>58200</td>
<td>45</td>
<td>1,5</td>
<td>105,107,117,120</td>
<td>16</td>
<td>18</td>
</tr>
<tr>
<td>122</td>
<td>58201</td>
<td>15</td>
<td>0,5</td>
<td>121</td>
<td>16</td>
<td>18</td>
</tr>
<tr>
<td>123</td>
<td>58250</td>
<td>10</td>
<td>0,5</td>
<td>122</td>
<td>16</td>
<td>19</td>
</tr>
<tr>
<td>124</td>
<td>58300</td>
<td>10</td>
<td>0</td>
<td>123</td>
<td>16</td>
<td>19</td>
</tr>
<tr>
<td>125</td>
<td>58310</td>
<td>20</td>
<td>1</td>
<td>124</td>
<td>16</td>
<td>19</td>
</tr>
<tr>
<td>126</td>
<td>58350</td>
<td>30</td>
<td>0,5</td>
<td>125</td>
<td>16</td>
<td>19</td>
</tr>
<tr>
<td>127</td>
<td>58351</td>
<td>10</td>
<td>0,5</td>
<td>126</td>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td>128</td>
<td>58400</td>
<td>25</td>
<td>0,5</td>
<td>117,120</td>
<td>14</td>
<td>18</td>
</tr>
<tr>
<td>129</td>
<td>58500</td>
<td>30</td>
<td>0,5</td>
<td>126</td>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td>130</td>
<td>58900</td>
<td>30</td>
<td>0,75</td>
<td>127,128,129</td>
<td>17</td>
<td>20</td>
</tr>
<tr>
<td>131</td>
<td>59000</td>
<td>40</td>
<td>0,5</td>
<td>117,120</td>
<td>14</td>
<td>19</td>
</tr>
<tr>
<td>132</td>
<td>59100</td>
<td>25</td>
<td>1</td>
<td>131</td>
<td>14</td>
<td>19</td>
</tr>
<tr>
<td>133</td>
<td>59300</td>
<td>25</td>
<td>0,5</td>
<td>130</td>
<td>17</td>
<td>20</td>
</tr>
<tr>
<td>134</td>
<td>59320</td>
<td>20</td>
<td>0,5</td>
<td>132</td>
<td>15</td>
<td>19</td>
</tr>
<tr>
<td>135</td>
<td>59340</td>
<td>15</td>
<td>0,5</td>
<td>134</td>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>136</td>
<td>59400</td>
<td>20</td>
<td>0,5</td>
<td>135</td>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>137</td>
<td>59500</td>
<td>30</td>
<td>0,5</td>
<td>136</td>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>138</td>
<td>59510</td>
<td>30</td>
<td>0,5</td>
<td>136</td>
<td>15</td>
<td>21</td>
</tr>
<tr>
<td>139</td>
<td>59600</td>
<td>15</td>
<td>1</td>
<td>137,138</td>
<td>15</td>
<td>21</td>
</tr>
<tr>
<td>140</td>
<td>59900</td>
<td>120</td>
<td>0</td>
<td>133,139</td>
<td>17</td>
<td>21</td>
</tr>
</tbody>
</table>

Instance from NISSAN Pathfinder Motor Engine Assembly Line Balancing, number (j), internal identifier (Id.), operation time (t), required area (a) and set of immediately predecessor tasks (P), are given for each task. Best found solution for SALBP-1 and TSALBP-1 are also reported, giving the station assignment of each task under the column SALBP-1 and TSALBP-1 respectively.

### References


