

To appear in the *International Journal of Production Research*
Vol. 00, No. 00, 00 Month 20XX, 1–21

A new model for supply chain network design with integrated assembly line balancing decisions

Jordi Pereira^{a*}, Mariona Vilà^b

^a*Faculty of Engineering and Sciences, Universidad Adolfo Ibáñez,
Av. Pedro Hurtado 750, Office C-216, Viña del Mar, Chile*

^b*Escola Universitària d'Enginyeria Tècnica Industrial de Barcelona (EUNETIB),
Universitat Politècnica de Catalunya. C. Comte Urgell, 187 1st. Floor, 08036 Barcelona, Spain*

(May 18, 2015)

Supply chain network design aims at the integration of the different actors of a supply chain within a single framework in order to optimize the total profit of the system. In this paper, we consider the integration of line balancing issues within the tactical decisions of the supply chain, and we offer a novel model and a solution approach for the problem. The new approach decomposes the problem into multiple line balancing problems and a mixed-integer linear model, which is easier to solve than the previously available nonlinear mixed-integer formulation. The results show that the new method is able to solve previously studied models within a fraction of the previously reported running times, and also allows us to solve larger instances than those reported by previous works. Finally, we also provide some analysis on the influence of the cost structure, the demand, and the structure of the assembly process on the final configuration of the assemblies and the distribution network.

Keywords: Supply Chain Design; Line Balancing; SALBP-1; MIxed Integer Linear Programming; Decomposition

1. Introduction

Supply Chain Network (SCN) design is a category of problems that has been widely studied in the last decades, see the reviews of Beamon (1998); Min and Zhou (2002); Mula et al. (2010) among others. The aim of SCN design is to coordinate the supply of parts, the manufacturing and the distribution of finished goods in a way that benefits suppliers, manufacturers and customers, **and it is an essential element of a successful supply chain.**

The design of the SCN includes decisions at both strategic and tactical levels **of supply chain management**, see Goetschalckx, Vidal and Dogan (2002). At the strategic level, we find decisions concerning the allocation and opening of facilities, and the design of the distribution network, whereas tactical decisions concern the manufacturing and warehousing of the products. Strategic decisions have a long-term effect on the structure of the supply chain network. As a consequence, tactical decisions are strongly dependent on strategic decisions, and there are multiple benefits in studying both types of decisions simultaneously, see Hall and Potts (2003).

In this paper, assembly line balancing is considered among the tactical decisions, and studied along with the design of the SCN. **The importance of studying assembly line balancing has been thoroughly highlighted (see Cachon and Terwiesch 2013, Chapter 4) as a method to improve productivity and reduce operation costs. Therefore, it is appropriate to consider line balancing within supply chain design in order to align pos-**

*Corresponding author. email address: jorge.pereira@uai.cl

sible savings within the SCN. The latter has been recently highlighted in Paksoy, Özceylan and Gökçen (2012); Hamta et al. (2015). Note that the operational level in decision-making, which corresponds to short-term decisions in the supply chain, is not considered within the SCN design problem.

An assembly line is a production system composed of a conveyor belt or a similar transportation device, which carries unfinished goods along several workstations, where the assembly process is carried out. The assembly process is divided in elemental tasks, which need a certain amount of time to be completed, and which may have precedence relationships among them (e.g., the doors of a car cannot be installed until the seats have been assembled). The assembly line balancing problem (ALBP) consists in finding an optimal assignment of tasks and/or resources to the workstations.

Many different balancing problems can be generated by considering different technical characteristics of the line, or by considering different objectives, see the reviews of Scholl and Becker (2006); Battaïa and Dolgui (2013). However, a substantial part of the literature regarding ALBPs is devoted to the basic formulation upon which the rest of the ALBPs are built. This basic formulation is known as the simple assembly line balancing problem (SALBP), and it is based on the following hypothesis, see Scholl and Becker (2006): (1) the mass production of a single homogeneous product; (2) a single, fixed and known production process; (3) a linear assembly line working at a constant pace (as a consequence, the amount of time available in each station, also known as the cycle time, is constant and equal for all of the stations); (4) a fixed, known, and deterministic duration for each task; (5) a constraint set corresponding to the precedence relationships between tasks; (6) serial and unilateral workstations; (7) independent stations, which are sufficiently equipped to carry out any of the tasks; and (8) an objective function which aims to maximize the efficiency of the line by minimizing the total idle time in all the stations.

Maximizing efficiency can be achieved by considering a fixed cycle time, c , and minimizing the number of stations, m (problem known as the SALBP-1), or by fixing the number of stations m and minimizing the cycle time c (problem known as the SALBP-2). A feasibility version of the problem also exists (known as SALBP-F), in which both the cycle time c and the number of stations m is known and the objective is to find a feasible solution. While the SALBP is a simplified version of the real ALBP problem, it constitutes the basic framework for real line balancing problems. **Note that any change in the SCN design (for instance, changes in the capacity of the facilities or the warehouses, changes in the demand or changes in the amounts transported) can inflict changes on the cycle time or the total number of available stations. This highlights the pertinence and the possible advantages of solving both the SCN design and the assembly line balancing problem jointly.**

Note that both the SALBP-1 and the SALBP-2 are known to be NP-hard, as they are a generalization of the Bin Packing Problem, which is known to be NP-hard, see Garey and Johnson (1979).

1.1 Literature review

Many different SCN design problems have been studied over the past decades, providing multiple formulations and resolution methods. The existing literature concerning SCN design problems is very diverse, as different authors include different decisions in their proposed models, as well as different objectives. Therefore, and due to the broad literature available on SCN design problems, we limit our study to those works on SCN with line balancing features, and we refer the reader to the literature contained in several recent reviews which we proceed to briefly describe. In Meixell and Gargeya (2005), the authors not only review the decision support models in the context of SCN design, but they also evaluate and critique the differences between the existing literature in this field and the actual problems found when designing a real-life SCN. A review of SCN design and management literature focused on environmental aspects can be found in Srivastava (2007). The application of SCN design in multiple different industries has been reviewed in Melo, Nickel and Saldanha-Da-Gama (2009), along with the performance measures used in the existing works in this

area. In Pishvae, Farahani and Dullaert (2010), the authors provide a critical review on SCN design under uncertainty, survey the different network design problems for forward, reverse and integrated supply chains, and propose a mixed-integer programming formulation for a bi-objective version of a SCN design problem. In Klibi, Martel and Guitouni (2010), the authors provide a critical review on SCN design under uncertainty, and also examine the different evaluation criteria used in SCN design, while Farahani et al. (2014) considers supply chain design subject to competition among different chains, and Farahani et al. (2015) considers location and inventory formulations on SCN.

Regarding assembly line balancing as part of the tactical decisions in SCN design problems is a research area only recently considered, see Paksoy, Özceylan and Gökçen (2012). Despite this, some works considered the coordination of the supply of parts and their assembly, see Sawik (2009), or applied ALBP technology to solve SCN design problems, see Che, Che and Hsu (2009).

To the best of our knowledge, Paksoy, Özceylan and Gökçen (2012) is the first existing study to combine SCN design with assembly line balancing decisions. This work proposes a non-linear, mixed-integer mathematical model to solve a multiple manufacturers, multiple assemblers, multiple customers problem, with a deterministic demand over multiple time periods. The objective is to minimize the total costs, taking into account the transportation costs and the fixed costs of the line (i.e., the cost of opening each station). The authors demonstrate the efficiency of the proposed formulation by means of a numerical example. Several scenarios were tested and sensitivity analyses were carried out to assess the influence of the parameters of the problem on the quality of the solutions obtained.

An extension for the previous model was developed in Paksoy and Özceylan (2012). In this study, a non-linear mixed-integer formulation is proposed for the SCN design problem, which includes U-shaped assembly line balancing decisions. The authors use a similar numerical example as the one used in the previously reviewed work to demonstrate the efficiency of the proposed model, and conclude that the simultaneous study of the SCN design and the U-shaped ALBP is beneficial.

More recently, two works that consider SCN design together with assembly line balancing decisions as well as uncertainty in the demand have been published. In Hamta et al. (2015), the authors propose a two-stage mixed-integer stochastic programming formulation for the resolution of the SCN design and ALBP with demand uncertainty. They also use several sampling strategy methods to obtain solutions for the largest instances more efficiently, and test the quality of the solutions of both the sampling strategies and the formulation using a set of randomly generated instances.

In Yolmeh and Salehi (2015), the authors propose a two-stage stochastic programming formulation for the exact resolution of the problem when considering demand uncertainty, and they later propose an Outer Approximation method to obtain good solutions for the larger instances in reduced running times. The Outer Approximation method is found to obtain better results for the practical-sized instances, both in terms of time and solution quality.

Assembly line design is an active research topic, which covers the different issues associated with the efficient configuration of the manufacturing process. In addition to the balancing of workload among stations, other authors have considered related problems. For example, in Emde and Boysen (2012), the authors study the routing and scheduling of the tow trains used to supply the stations of a mixed-model assembly line, solving both problems jointly using an exact algorithm. In the study by Limère et al. (2012), a mathematical cost model is presented to aid decision-making in material supply by comparing two supply systems: line stocking and kitting. A more recent study on the subject of kitting can be found in Faccio et al. (2015), in which the authors analyze different packaging strategies and then create a comparison model with the aim to aid operation managers with their decision-making. Another recent work, see Sternatz (2015), studies the interdependences between the assembly line balancing problem and the material supply problem, and develops a model in order to solve both problems jointly.

However, in this work we only consider the line balancing part of the assembly line design problem. Thus, the ALBP acts as a simplification of the problem in which

the number of workstations (the SALBP-1 objective function) is a measure of the manufacturing costs. Two excellent reviews of the state-of-the-art on ALBP can be found in Scholl and Becker (2006) and Battaïa and Dolgui (2013), and we refer the interested reader to their reviews.

Due to their use in the solution procedure proposed in Section 3, we also provide a succinct review on the state-of-the-art resolution procedures for the SALBP. The literature covers multiple approaches to solve the SALBP, including both heuristic, see Bautista and Pereira (2009); Pape (2015), and exact, see Sewell and Jacobson (2012); Vilà and Pereira (2013); Morrison, Sewell and Jacobson (2014), methods. For the SALBP-1, exact methods are based on enumeration, with branch-and-bound-based techniques being the preferred approach. This is due to the availability of tight lower bounds, Pereira (2015), and the ability to optimally solve instances with up to 1000 tasks within reduced running times. Moreover, the SALBP-1 methods can be applied to the SALBP-2, as the evaluation of the minimum cycle time c for a given number of stations m can be solved by iteratively solving the SALBP-1 with multiple tentative cycle times.

1.2 Contributions of this work

In this work, we propose an alternative formulation of the joint supply chain network design and assembly line balancing problem. The formulation is based on some observations, see Section 2, which allow us to decompose the problem into a sequence of type-2 simple assembly line balancing problems and a mixed-integer linear programming model. The mixed-integer linear program considers the transportation and represents the line balancing costs using a piecewise function obtained by the sequence of line balancing problems.

This approach is made possible by the availability of effective specific-purpose solvers for the assembly line balancing problem. While the simple assembly line balancing problem is NP-hard, it can be efficiently solved for large size instances, for example Morrison, Sewell and Jacobson (2014) reports the optimal resolution of instances with up to 1,000 jobs. Consequently, the resulting decomposition is able to solve the instance considered in Paksoy, Özceylan and Gökçen (2012) within a second, while, according to this study, the time required to solve the original formulation was 2 minutes and 20 seconds.

In addition to its applicability on the problem in hand, the decomposition approach is applicable to other integrated supply network design and assembly line balancing problems, provided that an effective algorithm exists for the resolution of the line balancing part of the problem, and the line balancing decision is linked to the distribution problem by the determination of the production level (units to assemble in each period).

A second contribution of this work is an analysis of the network costs, the demand structures and the assembly operations on the final structure of the solution. While Paksoy, Özceylan and Gökçen (2012) also considers different situations and their relation on the final network configuration, it does not analyze the relation among costs. In this paper we conduct a thorough study of the topic, including multiple different product structures (represented by different line balancing graphs).

1.3 Paper outline

The rest of the paper is divided as follows. Section 2 is devoted to the proposed decomposition method and documents the algorithmic contributions of this paper. Section 3 documents the computational experiment performed to assess the quality of the proposed method in terms of its applicability to solve large instances. In Section 4, we examine the results and analyze the influence of different characteristics of the instance on the final configuration of the network, **while Section 5 considers the advantages of the joint resolution of the line balancing and the network design problems by comparing the solutions obtained on a numerical example**

that acts as a case study. Finally, Section 6 provides some conclusions and future lines of work.

2. Problem formulation

In this Section we consider the original mathematical formulation of the problem as well as the proposed reformulation. Subsection 2.1 is devoted to the original formulation of the problem, and Subsection 2.2 puts forward the decomposition approach.

2.1 A mathematical formulation for the joint supply chain network design and assembly line balancing problem

In order to fully understand the problem under study and the proposed decomposition approach, we describe the mathematical model from Paksoy, Özceylan and Gökçen (2012) with some slight variations. The model uses the following parameters:

M : set of manufacturers ($m \in M$).

A : set of assemblers ($a \in A$).

C : set of customers ($c \in C$).

P : set of periods ($p \in P$).

K : set of components ($k \in K$).

J : ordered set of stations (the number of tasks can be used as an upper bound on the number of stations).

N : set of tasks ($i, r, s \in N$).

L : Set of precedence relations. $(r, s) \in L$ corresponds to a precedence relation between tasks r and s (r is a predecessor of s).

t_i : processing time of task i (time units).

WT_p : working time in period p (time units).

a_{mkp} : capacity of manufacturer m for component k in period p (units).

b_{ap} : capacity of assembler a in period p (units).

u_{cp} : demand of customer c in period p (units).

C_{map} : unit cost of shipping from manufacturer m to assembler a in period p (monetary units / distance units per unit).

C_{acp} : unit cost of shipping from assembler a to customer c in period p (monetary units / distance units per unit).

D_{ma} : distance between manufacturer m and assembler a (distance units).

D_{ac} : distance between assembler a and customer c (distance units).

O : fixed cost of opening a station in the assembly line (monetary units).

As the objective is to determine the network design as well as the assembly line design, variables regarding the total number of units shipped from the manufacturers to the assemblers, and from the assemblers to the customers, as well as variables regarding the assignment of the tasks to the stations are required. The variables of the model follow.

X_{makp} : units shipped from manufacturer m to assembler a for component k in period p .

Y_{acp} : units shipped from assembler a to customer c in period p .

V_{aijp} : binary variable which takes value 1 if assembler a performs task i in workstation j in period p , and 0 otherwise.

Z_{ajp} : binary variable which takes value 1 if there is a task assigned to station j for assembler a in period p .

CT_{ap} : cycle time for assembler a in period p .

According to the previous notation, the supply chain network design problem with integrated assembly line balancing can be stated using (1)-(14).

$$[\text{MIN}]Z_1 + Z_2 \quad (1)$$

subject to:

$$Z_1 = \sum_{m=1}^M \sum_{a \in A} \sum_{k \in K} \sum_{p \in P} D_{ma} C_{map} X_{makp} + \sum_{a \in A} \sum_{c \in C} \sum_{p \in P} D_{ac} C_{acp} Y_{acp} \quad (2)$$

$$Z_2 = O \sum_{a \in A} \sum_{j \in J} \sum_{p \in P} Z_{ajp} \quad (3)$$

$$\sum_{a \in A} X_{makp} \leq a_{mkp} \quad \forall m \in M, \forall k \in K, \forall p \in P \quad (4)$$

$$\sum_{c \in C} Y_{acp} \leq b_{ap} \quad \forall a \in A, p \in P \quad (5)$$

$$\sum_{a \in A} Y_{acp} \geq u_{cp} \quad \forall c \in C, \forall p \in P \quad (6)$$

$$\sum_{m \in M} X_{makp} - \sum_{c \in C} Y_{acp} = 0 \quad \forall a \in A, \forall k \in K, \forall p \in P \quad (7)$$

$$\sum_{j \in J} V_{aijp} = 1 \quad \forall a \in A, \forall i \in N, \forall p \in P \quad (8)$$

$$\sum_{j \in J} j V_{arjp} \leq \sum_{j \in J} j V_{asjp} \quad \forall a \in A, \forall (r, s) \in L, \forall p \in P \quad (9)$$

$$\sum_{i \in N} t_i V_{aijp} \leq CT_{ap} \quad \forall a \in A, \forall j \in J, \forall p \in P \quad (10)$$

$$\sum_{i \in N} V_{aijp} - |N| Z_{ajp} \leq 0 \quad \forall a \in A, \forall j \in J, \forall p \in P \quad (11)$$

$$CT_{ap} = \frac{WT_p}{\sum_{c \in C} Y_{acp}} \quad \forall a \in A, \forall p \in P \quad (12)$$

$$X_{makp}, Y_{acp}, CT_{ap} \geq 0 \quad \forall m \in M, a \in A, \forall k \in K, \forall c \in C, \forall p \in P \quad (13)$$

$$V_{aijp}, Z_{ajp} \in \{0, 1\} \quad \forall a \in A, \forall i \in N, \forall j \in J, \forall p \in P \quad (14)$$

Equation (1) indicates that the objective is to minimize the sum of the transportation costs, Z_1 calculated using (2), and the costs of the assemblers, Z_2 calculated using (3). **Note that, while a multiobjective approach could be used, this work opts for an integrated view of the supply chain optimization process, as seen in Chopra and Meindl (2012) among others. Consequently, Z_1 and Z_2 are measures of the transportation and manufacturing costs respectively, and their cost coefficients in (2) and (3) should represent their relative importance in the total cost of the supply chain.**

The constraints correspond to: a) the formulation of the network design problem, equations (4)-(7), b) the formulation of the assembly line balancing problem, (7)-(11), c) some linking constraints between both problems, constraint set (12), and d) the definition of the domain of the variables, (13)-(14).

Constraint sets (4)-(7) correspond to the constraints of a multi-commodity network flow problem in which different components flow from a set of sources (the manufacturers) to the intermediate nodes (the assemblers, in which the components are joined into a single final product), and then are sent to the final destination (the customers). Constraint sets (4) and (5) make sure that the capacity of the manufacturers and the assemblers is satisfied, while constraint set (6) verifies that the demand of each customer is satisfied. Finally, constraint set (7) corresponds to the flow constraints in each assembler, and also ensures that each final product contains each of the required components.

The line balancing constraints correspond to constraint sets (8)-(11); constraint set (8) ensures that every task is assigned to one station in each assembler and time period, constraint set (9) represents the precedence constraint among tasks; and (10) enforces the cycle time constraint (the tasks assigned to a station never surpass the available cycle time). Finally, constraint set (11) calculates whether a station is open; that is, if any task has been assigned to station j in assembler a during period p , then Z_{ajp} must be equal to 1. Note that there is a difference between the model proposed in Paksoy, Özceylan and Gökçen (2012) and the our constraint set. Precedence constraints, constraint set (9), have been corrected from the original description. The original model did not contain the multiplication by the station number (multiply each term of the summation by its corresponding station number j), which leads to a set of redundant constraints (any task assignment fulfils the constraint set).

Finally, constraint set (12) joins both problems by linking the flow of units through the assembler and the production rate.

2.2 A decomposition-based formulation of the problem

While the previous model is able to handle small-sized instances to optimality (previous results report the resolution of an instance with up to 8 tasks), it faces two issues that hamper its ability to tackle large-size problems: (1) constraint set (12) is non linear; and (2) simple assembly line balancing problems, objective (3) and constraint sets (8)-(11), are challenging problems for general-purpose solvers.

In order to avoid the non-linearity of the model, as well as to use a special purpose solver to

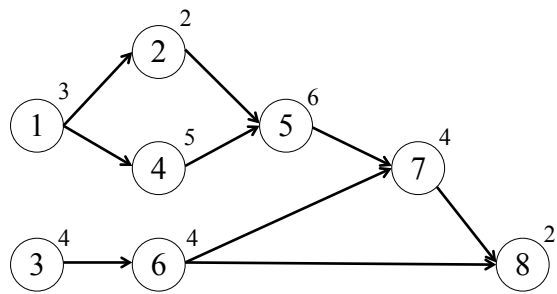


Figure 1. SALBP-1 instance from Paksoy, Özceylan and Gökçen (2012). Each task is represented by a vertex with its processing time in the upper right corner of the vertex, and arcs representing the precedence relations.

obtain solutions for the assembly line balancing formulation, we propose to divide the problem into the resolution of a sequence of assembly line balancing problems, and a mixed-integer linear program. The decomposition is based on the following observations:

Observation 2.1. *For any given production level on an assembler on any given period, $Y_{ap} = \sum_{c \in C} Y_{acp}$, there is an optimal number of stations which does not depend on the distribution network.*

Observation 2.1 is a straightforward interpretation of equation set (12), which links the production level to the cycle time of the assembler, and the existence of an optimal solution for its corresponding SALBP-1.

Observation 2.2. *The number of stations required by an assembler for any given period is a non-decreasing function with respect to the production level.*

Again, Observation 2.2 is a straightforward interpretation of the relationship between the number of stations of an assembly line and its cycle time. If the production level increases (or equivalently, the cycle time decreases) the optimal number of stations increases or remains the same.

In order to determine the optimal number of workstations for any production rate (cycle time), we only need to solve at most $|N| - 1$ SALBP-2 instances (that is, find the minimum cycle time for which a solution with a given number of workstations exist) to define the range of production rates which can be accomplished with a given number of stations. The previous observation indicates that there exists a piecewise function which links any production level, Y_{ap} , with the optimal number of stations of the assembler (the value used to calculate the assembly costs in (3)).

We will illustrate the aforementioned observations using the assembly line example provided in Paksoy, Özceylan and Gökçen (2012). The example has eight tasks with the precedence diagram depicted in Figure 1, which includes the task times, in minutes, on the upper right side of each vertex. The total working time for any assembler during any period is equal to 28,800 minutes.

If no units are produced in an assembler during period p , $Y_{ap} = 0$, the cost of the assembler during this period is equal to 0.

In order to determine the maximum production rate for which it is sufficient to use a single workstation, we calculate the optimal solution for the SALBP-2 with 1 workstation. Its value is equal to $\sum_{i \in N} t_i = 30$. Consequently, if the production rate $Y_{ap} \leq 28,800/30 = 960$, the total assembly cost is equal to the constant cost associated to opening a station, O .

For two stations, the optimal cycle time is 16. Therefore if $960 < Y_{ap} \leq 28,800/16 = 1800$ units are produced, the optimal number of workstations is 2, and the assembly cost is $2 \cdot O$.

If we proceed and solve the SALBP-2 instances with 3, 4, 5, 6, 7 and 8 stations, a characterization of the cost associated to different production rates is obtained. Table 1 reports the results for the example given in Figure 1. Note that using 7 or more workstations would not reduce the cycle time, as the tasks are not divisible, and the cycle time is equal to the maximum processing time among the tasks, see SALBP hypothesis in Section 1.1. Hence, there is no need to consider the resolution of problems with more workstations than the number of tasks.

Table 1. Solution to the example instance from figure 1.

Stations	Cycle time	Production rate
0	-	0
1	30	(0,960]
2	16	(960,1800]
3	10	(1800,2880]
4	8	(2880,3600]
5	7	(3600,4114.29]
6	6	(4114.29,4800]

The intervals from Table 1 correspond to a piecewise structure with discontinuities. Moreover, Observation 2.2 combined with the minimization objective allows us to model the cost structure with the use of an additional set of binary variables. Let Z'_{ajp} be a binary variable which takes value 1 if j stations are opened in assembler a during period p . For the example given in Paksoy, Özceylan and Gökçen (2012), the assembly costs Z_2 can be represented by equation (15) and constraints (16) and (17).

$$Z_2 = \sum_{a \in A} \sum_{p \in P} \sum_{j \in J} O \cdot j \cdot Z'_{ajp} \quad (15)$$

$$Y_{ap} \leq 960Z'_{a1p} + 1800Z'_{a2p} + 2880Z'_{a3p} + 3600Z'_{a4p} + 4114.29Z'_{a5p} + 4800Z'_{a6p} \quad \forall a \in A, \forall p \in P \quad (16)$$

$$\sum_{j \in J} Z'_{ajp} \leq 1 \quad \forall a \in A, \forall p \in P \quad (17)$$

We now proceed to generalize the previous results. Let $|J|$ be the optimal number of workstations required to perform all of the assembly tasks with a cycle time equal to $\max_{i \in N} t_i$, let \bar{u}_j be the upper bound on the production level, Y_{ap} , reported by the optimal solution of the SALBP-2 problem with a given number of stations j , and let Z'_{ajp} be a set of binary variables which take value 1 if j stations are opened in assembler a during period p . Then Z_2 and the assembly line balancing part of the original model, constraint sets (8)-(13), can be rewritten as follows:

$$Z_2 = \sum_{a \in A} \sum_{p \in P} \sum_{j \in J} O \cdot j \cdot Z'_{ajp} \quad (18)$$

$$Y_{ap} \leq \sum_{j \in J} \bar{u}_j Z'_{ajp} \quad \forall a \in A, \forall p \in P \quad (19)$$

$$\sum_{j \in J} Z'_{ajp} \leq 1 \quad \forall a \in A, \forall p \in P \quad (20)$$

Equation (18) and constraints sets (19) and (20) can be easily integrated within commercial linear programming solvers like CPLEX (see <http://www.ibm.com>) or Gurobi (see <http://www.gurobi.com>). Furthermore, the function encapsulates the line balancing decision, and provides the optimal number of stations for any given production rate, basically enabling the decomposition of the problem into several SALBP-2 and a mixed-integer linear programming model

constituted by objective (1), whose costs can be calculated using (2) and (18), the constraint sets (4)-(7), (19), (20), and the definition of the domains for the variables.

The previous decomposition requires the resolution of multiple SALBP-2 instances. Although the SALBP-2 is NP-hard, and thus we could opt for its resolution using a heuristic approach, it is possible to optimally solve large-sized instances due to its relation to the SALBP-1 and the existence of several special-purpose procedures for the task in hand (the literature reports that SALBP-1 instances with up to 1,000 tasks have been solved to optimality within reduced running times).

Consequently, the SALBP-2 instances are solved using an iterated search method in which we successively solve instances of the SALBP-1 with different trial cycle times until the minimum cycle time with the desired number of workstations is found. This approach has been proposed and tested in several works, see Scholl and Becker (2006), and provides an efficient method to solve the problem which can make use of the currently available procedures for the SALBP-1, like Vilà and Pereira (2013); Morrison, Sewell and Jacobson (2014). In this case, and in order to facilitate the reproducibility of the results, we use the code from Morrison, Sewell and Jacobson (2014) as it is publicly available (see <http://www.assembly-line-balancing.de>). As multiple SALBP-2 instances are to be solved, the search is structured as follows, so as to avoid work repetition:

- We first solve the SALBP-1 with cycle time $CT = \max_{i \in N} t_i$. This value provides an upper bound on the production rate, the maximum number of stations $|J|$.
- The optimal cycle time for 1 station corresponds to the sum of processing times of all of the tasks, that is, $CT = \sum_{i \in N} t_i$.
- For the remaining SALBP-2 instances, with number of stations ranging from 2 to $|J| - 1$, we solve the problem using a binary search approach, see Scholl and Becker (2006). The binary search first defines a lower, LC , and an upper, UC , bound on the optimal cycle time. Then, the instance with tentative cycle time equal to $CT' = \lfloor (LC + UC)/2 \rfloor$ is solved to optimality. If the optimal number of stations for CT' is larger than or equal to the desired number of workstations UC is updated ($UC = CT'$), otherwise, LC is updated ($LC = CT' + 1$). The initial values for the interval of any given number of stations j are obtained as follows: LC corresponds to a lower bound on the cycle time, $\lceil \sum_{i \in N} t_i / j \rceil$, or the maximum processing time of any task, $\text{MAX}_{\forall i \in N} t_i$; UC corresponds to the minimum among the total processing time of all of the tasks $\sum_{\forall i \in N} t_i$, or the smallest cycle time of any previously solved problem in which the number of stations is smaller than or equal to j .

Like in the previous example, once all of the cycle times are known, we obtain the maximum production level, \bar{u}_j , for any given number of workstations, j , by dividing the total working time of the period by the minimum cycle time for j workstations.

3. Computational Results

In order to evaluate the quality of the proposed method, we solved the instance from the case study proposed in Paksoy, Özceylan and Gökçen (2012), and the optimal solution was found in less than 1 second, including the time required to optimally solve the SALBP-2 instances. Therefore, we conducted a larger computational experiment on a set of randomly generated instances in order to evaluate the quality of the proposed method and the impact of different configurations on the assembly and network design decisions. These instances are constructed in such a way that: (1) different degrees of occupation level and cost ratio are considered; (2) the cost of the joint network design and assembly balancing decisions can be compared to the cost of taking the same decisions hierarchically; and (3) different assembly structures are considered.

All of the computational experiments were run on an Intel Xeon machine with an 8-core 2.66 Ghz CPU and 32-GB RAM running the Linux operating system. CPLEX was used to solve the MILP models and the branch, bound and remember code available at <http://www.assembly-line->

balancing.de was used to solve the SALBP-1 instances. All the remaining parts of the proposed methods were coded in C++. The code uses a single thread in order to perform all of the operations, including CPLEX, whose maximum number of CPUs in use was set to 1. Furthermore, we allow 600 seconds of computing time for CPLEX. The limit was set given the fact that in a preliminary computational experiment, CPLEX failed to verify the optimality of the incumbent solution in some instances, even when very small optimality gaps (below 0.1%) were reported.

The remainder of the section describes the instances and studies the performance of the proposed algorithm. The analysis of the results in terms of the impact of jointly solving network design and assembly line balancing is the focus of Section 4.

3.1 Description of the instances

In order to generate instances which can be used to analyze the quality of the presented procedure, a structured approach was used for the construction of the instance set.

Definition of the assembly line:

- The instances with 50 tasks from the dataset proposed in Otto, Otto and Scholl (2013) are used, consequently $|N| = 50$. These instances contain different characteristics regarding their processing time distribution and the structure of the precedence relations, which we proceed to describe (see Otto, Otto and Scholl (2013) for more details):
 - Three different processing time structures are considered: *bottom*, in which the processing time of the tasks are generated using a normal distribution with a peak at 100; *medium*, in which the peak is set at 500; and *bimodal*, in which a bimodal distribution with peaks at 100 and 500 time units is used
 - Three different order strength structures (ratio between the number of precedence relations in the instance and the maximum number of precedence relations) are considered: *low*, in which the ratio is equal to 0.2; *medium*, with a ratio equal to 0.5 and *high*, with a ratio equal to 0.9
 - Three different precedence constraints structures are also considered: *block*, in which there are some tasks with multiple predecessors and successors; *chain*, in which there are many tasks having a predecessor and a successor; and *mixed*, in which both chain and block tasks appear
- The set of predecessors L , and processing time of the tasks, $t_i \forall i \in N$, corresponds to the processing times indicated in the instances proposed in Otto, Otto and Scholl (2013). Processing times can take values ranging from 1 to 1,000 time units. We interpret a time unit as a centesimal part of a minute, hence the processing time of the tasks range from slightly less than a second to 10 minutes. The time unit value was chosen in order to provide task processing times similar to those in Paksoy, Özceylan and Gökçen (2012).
- The maximum number of stations, $|J|$, is set to the optimal solution for the SALBP-1 instance with cycle time equal to $\text{MAX}_{\forall i \in N} t_i$. Note that additional stations would not provide any benefit to the production system.
- The number of assemblers is set to the value of the largest instance reported in Hamta et al. (2015). Therefore, $|A| = 22$.
- The working time per period, WT_p , is set to 28,800 minutes, as in Paksoy, Özceylan and Gökçen (2012), and the capacity of the assembler, b_{ap} is set to the maximum possible production level of the assembly according to the assembly line balancing problem. The maximum production level is related to the minimum cycle time of the assembly line which corresponds to the largest processing time among the tasks ($c_{min} = \text{MAX}_{\forall i \in N} t_i$). Consequently, the maximum production level per period and assembler is $100WT_p/c_{min}$. Therefore, b_{ap} is a constant b for all assemblers and periods.

Definition of the distribution network:

- The number of components, customers, manufacturers and periods is set to the values of the largest instance reported in Hamta et al. (2015). Therefore, $|K| = 10$, $|C| = 25$, $|M| = 20$ and $|P| = 2$.
- The costs of shipping from each manufacturer to each assembler, C_{map} , and from each assembler to each customer, C_{acp} , are drawn from a uniform distribution, $\sim U(0.03, 0.08)$, as in Hamta et al. (2015) **(the costs were scaled down to improve numerical stability)**.
- The distances between each manufacturer and assembler, D_{ma} , and between each assembler and customer, D_{ac} , are drawn from a uniform distribution, $\sim U(200, 600)$, as in Hamta et al. (2015).
- Three different demand levels for each customer and period, u_{cp} , are considered. These levels are used to evaluate the impact caused by the capacity of the manufacturers on the structure of the final supply network. Three levels of demand (*low*, *medium*, *high*) are established in accordance to a desired ratio between the maximum production level of the assemblers and the demand of the customers. As the maximum production level for each assembler and period has been set to b , the demand of each customer and period is calculated as follows:
 - If a *low* ratio between demand and maximum throughput is considered (25%), the average demand of each customer is $\hat{d} = \frac{0.25b|A|}{|C|}$. The final demand of each customer per period is then randomly drawn from a uniform distribution, $\sim U(0.9\hat{d}, 1.1\hat{d})$.
 - If a *medium* ratio between demand and maximum throughput is considered (50%), the average demand of each customer is $\hat{d} = \frac{0.5b|A|}{|C|}$. The final demand of each customer per period is then drawn from a uniform distribution, $\sim U(0.9\hat{d}, 1.1\hat{d})$.
 - If a *high* ratio between demand and maximum throughput is considered (75%), the average demand of each customer is $\hat{d} = \frac{0.75b|A|}{|C|}$. The demand of each customer per period is randomly drawn from a uniform distribution, $\sim U(0.9\hat{d}, 1.1\hat{d})$.
- Manufacturers capacity, a_{mkp} , was set to approximately 4.5 times the average demand of the customers, as in Hamta et al. (2015). As the expected total demand for any period is $\hat{D} = \hat{d} \cdot |C|$, the average capacity of each of the manufacturers is $4.5\hat{D}/|M|$, and the capacity of each manufacturer for each component and period is drawn from a uniform distribution, $\sim U(0.9 \cdot 4.5 \cdot \hat{D}/|M|, 1.1 \cdot 4.5 \cdot \hat{D}/|M|)$.

The generation of assembly costs is performed in such a way that allows us to quantify the difference between solving the integrated problem and optimally solving both parts of the problem sequentially (i.e., optimally designing the supply network, to later solve the assembly balancing problem to optimality based on the production level of each assembler). Therefore, three different levels of assembly costs (*low*, *medium* and *high*) are considered, which are used to generate the assembly costs as follows:

- We first obtain the optimal cost of the distribution network disregarding the assembly costs. The optimal cost of the distribution network corresponds to the resolution of the multicommodity flow model with objective Z_1 , equation (2), and constraint sets (4)-(7).
- Then, the optimal (minimum) number of workstations in each assembler and period required to reach the production levels for the optimal distribution network are ascertained. The number of workstations corresponds to the optimal solution of the SALBP-1 instances, using the cycle time defined by the production rate of each assembler at each time period. Let W be the total number of workstations.
- Finally, let us consider the total cost be $Z_1 + O \cdot W$. Then, we set O , the cost associated to opening each station, as follows:
 - If the assembly cost level is set to *low*, the total assembly costs are set to be equal to 10% of the total cost. Consequently, $O = \frac{Z_1}{9W}$
 - If the assembly cost level is set to *medium*, the total assembly costs are set to be equal to 30% of the total cost, and thus $O = \frac{3Z_1}{7W}$
 - If the assembly cost level is set to *high*, the total assembly costs are set to be equal to

Table 2. Average running time (in seconds) required to solve the assembly line balancing part (column SALBP) and the mathematical formulation (column MILP) of the proposed method.

Structure	OS	times	av. t (SALBP)	av. t (MILP)
block	low	bottom	5554	13.9
		middle	414.7	85.8
		bimodal	397.2	81.5
	medium	bottom	543.4	15
		middle	148.5	53.9
		bimodal	125.7	59.8
chain	low	bottom	6400.9	19.5
		middle	198.7	68.9
		bimodal	246.1	63.3
	medium	bottom	555.4	14.5
		middle	168.2	43.7
		bimodal	124.6	52.7
mixed	low	bottom	6283.5	17.2
		middle	434.3	83
		bimodal	498.7	58.6
	medium	bottom	500.1	14.1
		middle	128.6	59.6
		bimodal	108.7	57.5
	high	bottom	568.3	5.9
		middle	186.6	24.9
		bimodal	174.1	21.2

60% of the total cost. Therefore, $O = \frac{3Z_1}{2W}$

Note that the fractions used to define O ensure that the total assembly cost, $O \cdot W$, is equal to 10% (*low*), 30% (*medium*) or 60% (*high*) of the total network cost if each problem is solved separately.

The aforementioned cost structure provides us with a method to study the improvements obtained by jointly considering the distribution and assembly line costs when compared to the separate resolution of the problems (first solving the network design problem and then considering the assembly line balancing problem). The transportation costs of the optimal joint solution are to increase, while the assembly line costs are to decrease when the joint problem is considered.

Note that while the instances were randomly generated, their parameters were based on previously published data that rely on real-life case studies, Hamta et al. (2015), interviews or previously published field studies, Otto, Otto and Scholl (2013). Hence, they can be considered as approximations to real life situations.

3.2 Results of the computational experiment

According to the previous generation scheme, a total of $525 \cdot 3 \cdot 3 = 4,725$ instances were generated. The performance of the proposed method is evaluated according to the average running time required to solve the instances. We do not report gaps between the upper and lower bounds provided by CPLEX after the 600 s time limit, since these values are very small (below 0.01% on average, and below 0.1% in all of the cases) and thus it can be considered that all of the instances can be solved to optimality.

Table 2 provides the average running time (wall time) among the instances when grouped according to their assembly characteristics (the SALBP part of the problem). Two values are provided: the time required to solve the SALBP-2 instances in order to encapsulate the line balancing decisions, column av. t (SALBP), and the time required to solve the mixed-integer linear program, column av. t (MILP). The average time is reported in seconds.

Table 3 provides the average running time among instances when grouped by their cost and demand characteristics. Table 3 does not report the time required to encapsulate the line balancing decisions, because it is equal for all of the combinations of manufacturer's capacity and demand level (all of the 525 alternative SALBP graphs are used once in each combination).

Table 3. Average running time (in seconds) required to solve the the mathematical formulation of the proposed method. Instances are grouped according to the assembler’s cost (rows) and the demand levels (columns). We only denote the time required to solve the mathematical formulation as the average time to solve the SALBP part is equal for all of the groups.

Demand level	assembler’s cost		
	low	medium	high
low	3.2	64.7	235.4
medium	2	9.1	57.4
high	1.5	2.1	16.6

The results show that the structure of the SALBP instances does not only affect the average time required to define the SALBP part of the proposed mathematical model, but also affects the time required to solve the corresponding mixed-integer linear programming model (MILP). While the average time required by the MILP model is relatively small (more so for instances with small processing times), the time required by the SALBP encapsulation may be prohibitively large (e.g. the average time for low order strength with bottom times instances exceeds one hour of computing time and its maximum is above 5 hours). For the remaining combination of instance characteristics, the average running times are approximately around the 10 minutes mark, which is a manageable running time.

Note that the previous results differ in the set of instances which are considered to be hard for SALBP algorithms, see Morrison, Sewell and Jacobson (2014); Pereira (2015), which indicates that instances with *medium* processing time are more difficult to solve. We conjecture that the cause of this perceived difficulty and the differences with previous results are derived from the resolution of different cycle times (as required by the SALBP-2).

The results in Morrison, Sewell and Jacobson (2014); Pereira (2015) were obtained for the SALBP-1 using a single cycle time ($c = 1000$). In this study we consider the resolution of multiple SALBP-1 with different cycle times. In this case, there are instances with *bottom* processing times in which most of the tasks will have a processing time with the characteristics of the *medium* distribution when $c = 1000$ (as it is the case when $c = \text{MAX}_{i \in N} t_i$). Nevertheless, the processing times of the tasks for instances with *bottom* distribution have a higher number of similar values than their middle distribution counterparts. Consequently, the resulting instances are more difficult to solve due to the inability of the algorithm to properly distinguish dominance relationships.

When we consider the characteristics of the assembler’s cost and the demand level, the results show that instances with *low* demand level and *high* manufacturer’s cost are ostensibly more difficult to solve than the remaining instances. The same behavior is also observed for all of the *high* assembly costs, and all of the *low* demand level instances. We conjecture that each phenomenon has a separate cause: (1) *high* assembler’s cost instances are more difficult to solve because the assembly decisions play a stronger role in the problem, and thus CPLEX has more trouble identifying the optimal number of stations; and (2) *low* demand level instances are more difficult to solve because there are more alternative methods to satisfy the components to the assemblers, and to move the final product from the assembler to the customer.

In short, the results show that the proposed method is a viable option to solve the integrated supply network design with assembly line balancing problem. Furthermore, for those cases in which the total running time required to solve the SALBP part of the problem is deemed excessive, we suggest the use of a heuristic procedure to identify the set of efficient solutions, we refer the reader to Section 6.

4. Influence of the different costs on the supply chain network

While the previous section identifies the ability of the proposed method to solve larger instances than those previously found in the literature, in this section we consider the effect of the different characteristics of the instances on the costs. Therefore, the following section tries to answer the following question: When is it convenient to integrate the line balancing decision on the supply chain network design?

In order to provide an answer to this question, we have analyzed the improvement of the objective value provided by the joint resolution of the supply chain network design problem and the assembly line balancing problem over the sequential resolution of these problems (that is, first solve the supply chain network design problem, and then optimally balance the assembly lines according to the desired production levels).

The optimal solution for the sequential problem corresponds to the optimal resolution of the multi-commodity flow model defined by objective function (2) and constraint sets (4)-(7) (the supply chain network design problem), and the resolution of the SALBP-1 instances defined by the production level for each assembler and period (the assembly line balancing part). Note that the total objective value of the problem is an upper bound on the optimal solution of the integrated model. Moreover, if we consider the contribution of each part of the problem to the total objective function and compare their contribution with the optimal costs for each of the sequential problems, the following observations hold:

- The cost of the supply chain network design problem in the sequential approach is smaller than or equal to the cost of the supply chain part of the integrated problem (Z_1).
- The cost of the assembly line balancing part of the problem in the sequential approach is higher or equal to the cost of the assembly line balancing part of the integrated problem (Z_2).

These observations can be derived from the relation between both problems: As the solution to the sequential approach is a feasible solution of the integrated problem, the total objective of the sequential solution can be equal to the objective of the integrated problem, but not smaller. Likewise, the objective function of the network design in the sequential approach corresponds to the optimal distribution network without assembly line considerations, and thus the optimal network for the integrated problem can only provide a greater or equal objective value. Therefore, if the total integrated objective value is to be smaller than the sequential objective value, the assembly line balancing part of the objective in the integrated problem must be smaller than the cost in the sequential approach.

In short, it is possible to improve the total solution by increasing the network design costs and reducing assembly costs, which we achieve by taking advantage of optimal production levels for a given number of stations.

Therefore, for each of the 4,725 instances considered in the computational experiment, we calculate the relative per cent difference, rd , between the optimal cost and the sequential cost, see (21), in which Z^* correspond to the optimal cost of the integrated problem, and Z^s to the sum of the optimal costs of the sequential problems.

$$rd = \frac{Z^s - Z^*}{Z^*} \cdot 100 \quad (21)$$

The maximum relative difference observed among all of the 4,725 instances is 16.9%, while the average value is 2.1%, indicating that the effect of considering the assembly line decision together with the supply network design may lead to significant cost reductions.

In order to verify which features are more significant on the variability of the cost reduction, Table 4 provides the average relative difference when instances are grouped according to their assembler's costs and demand levels.

The results show that the most important factor is the assembly cost. If the assembly cost is

Table 4. Average relative per cent difference (rd) between the sequential and the integrated costs. Instances are grouped according to the assembler's cost (columns) and the demand levels (rows).

demand level	assembler's cost		
	low	medium	high
low	0.335	1.944	5.27
medium	0.204	1.449	4.844
high	0.128	0.96	3.769

Table 5. Average relative per cent difference (rd) between the sequential and the integrated approaches. Instances are grouped according to the characteristics of their respective SALBP instance).

Structure	OS	processing times		
		bottom	middle	bimodal
block	low	4.86	0.764	0.969
block	medium	4.111	0.85	0.966
chain	low	4.639	0.764	0.693
chain	medium	4.526	0.869	0.949
mixed	low	4.485	0.657	0.813
mixed	medium	4.485	0.901	0.906
mixed	high	3.869	1.608	1.423

high (column *high* of the assembler's costs), the benefits of the joint resolution of the problem are larger. While the effect of the demand level (rows) is still significant, it is smaller, indicating that the improvement on supply chains with low demand levels (with respect to the total available capacity) is more significant. We conjecture that this is because instances with high demand level require a larger number of assemblers to use their maximum assembly capacity, both in the sequential and the integrated solution approaches, which leads to less significant differences between them.

Table 5 provides the relative per cent difference (rd) when the instances are grouped according to the characteristics of the assembly operations (their respective SALBP instance).

Among the instance characteristics, only the processing times seem to significantly affect the relative difference between the sequential and the integrated approach. The effect of integrating the line balancing decision on instances with *bottom* processing time distribution is more noticeable. We infer that this is caused by the larger possibilities to obtain solutions with fewer stations when the instance contains multiple operations with small processing times (in these instances it is easier to find stations assignments with smaller idle times).

Consequently, the results lead us to conclude that while some benefits exist in jointly considering the supply chain network design problem with the assembly line balancing problem, the largest improvements are obtained when the following characteristics arise: a) the assembler cost plays an important role, when b) it is possible to decide on different assemblers (*low* demand), and c) when the assembly requires the execution of multiple small tasks. In those cases, large improvements (around 4% in average and up to 16.9% in one of the considered instances) can be achieved by jointly considering both decisions.

5. Influence of integrating network design and line balancing decisions on the structure of the supply chain

In this section we analyze the advantages of the joint resolution of the problem over the design of the supply network followed by the line balancing decisions. The difference between both approaches is investigated by comparing the solutions obtained by the proposed method and the sequential resolution of the problem on a specific instance that acts as a case study.

Table 6. Transportation costs (column Z_1), assembly costs (column Z_2) and total costs (column $Total$) obtained on the case study by solving the SCN design and line balancing problems sequentially (row Sequential) and jointly (column Joint).

	Z_1	Z_2	$Total$
Sequential	7857468 (70%)	3355562 (30%)	11213030
Joint	8124113 (76%)	2565607 (24%)	10689720

Table 7. For each resolution method (rows Sequential and Joint), the average number of open manufacturers and assemblers per period is given (column #), as well as the average number of transportation links from manufacturers to assemblers and from assemblers to costumers per period (columns Edges M-A and Edges A-C).

	Manufacturers		Assemblers	
	#	Links M-A	#	Links A-C
Sequential	12	12	12	35.5
Joint	15.5	15.5	15.5	39.5

The instance selected has the same network structure as the biggest instances in Hamta et al. (2015) (as such, the number of assemblers, components, manufacturers, customers and periods is the same as in the instances solved in Section 3). The assembly line balancing instance used corresponds to instance n=50_33 in Otto, Otto and Scholl (2013). The order strength of the instance is *low* (few precedence constraints), the processing times structure is *bottom* (multiple tasks with small processing times) and the structure of precedence constraints is *mixed* (hence, the instance contains both *chain* and *block* precedence constraints).

To avoid the analysis of an extreme case, average parameters were chosen for generating the instance (with the exception of the *bottom* processing time structure). Consequently, the assembly cost level and the ratio between the demand and the maximum throughput were set to *medium* to create the network. An instance with *bottom* processing times was selected because this characteristic identifies the set of instances in which the integrated resolution offers the largest opportunities for improvement according to the results shown in Section 4 (Table 5). Please note that while the instance under study does not correspond to a real-life situation, both the supply chain network design information and the line balancing information were based on previously published case studies, interviews and field studies, see Hamta et al. (2015); Otto, Otto and Scholl (2013).

Table 6 details the differences in the objective function for the sequential and joint resolution of the case study. The previous section analyzed in more detail the influence of the joint resolution on the costs. However, it is important to note that the joint resolution of these problems decreased the total costs, by slightly increasing the transportation costs and substantially decreasing the assembly costs. The joint solution provides savings of a 4.7% over the sequential solution.

To analyze the effect that the joint resolution has on the distribution network, Table 7 provides different metrics associated to the paths taken by the components from the manufacturers to the assemblers, as well as the paths taken by the products from the assemblers to the customers.

For each resolution method (be it sequential or joint) the average number of manufacturers and assemblers per period with positive production level (column #) and the average number of transportation links used between manufacturers and assemblers and between assemblers and customers per period (Edges M-A and Edges A-C) are given.

Table 7 shows that approximately half the manufacturers and assemblers are used in both solutions, with a clear increase in the number of open assemblers in the joint solution (due to the smaller production level per assembler, see Table 9). Furthermore, each assembler is supplied by a single manufacturer, while each customer is usually supplied by different assemblers.

Table 8 includes metrics regarding the average characteristics of the links used in the transportation network. It reports the minimum, the maximum, the average and the standard deviation of

Table 8. For each resolution method (rows Sequential and Joint), and type of transportation link (manufacturer to assembler, M-A, and assembler to customer, A-C), the minimum (column min), maximum (column max), average (column av) and standard deviation (column sd) is given for the following metrics: the number of units sent; the transportation cost per unit sent; and the distance covered per unit sent.

	Units sent M-A				Units sent A-C			
	min.	av.	max.	sd.	min.	av.	max.	sd.
Sequential	3870	36300	39560	9346	32	1227	1886	594
Joint	2770	28103	30700	6029	65	1102	1882	550

	Transportation cost per unit M-A				Transportation cost per unit A-C			
	min.	av.	max.	sd.	min.	av.	max.	sd.
Sequential	6.38	7.49	9.21	0.72	6.28	11.67	18.38	2.82
Joint	7.08	8.35	10.54	0.91	6.28	11.26	18.53	2.69

	Distance M-A				Distance A-C			
	min.	av.	max.	sd.	min.	av.	max.	sd.
Sequential	201.919	223.387	262.290	16.608	201.901	282.576	434.073	56.978
Joint	201.919	230.403	340.131	9.106	201.901	279.029	440.464	55.190

the number of units sent between manufacturer and assembler or assembler and customer, their respective transportation cost and the distance covered per unit. Note that the total number of units sent from manufacturer to assembler is ten times larger than from assembler to customer (as ten components are assembled into a single product). The results shown in Table 8 highlight the following:

- (1) A larger amount of components is sent between a manufacturer and an assembler in the sequential solution than in the joint solution. This result corresponds to the logical optimization of the transportation costs in the sequential model, which leads to higher volumes of components being sent through more efficient links.
- (2) As the number of components that need to be moved is larger than their corresponding products, the sequential resolution obtains a much more efficient manufacturer-assembler network than a assemble-customer network.
- (3) The joint solution offers better metrics per unit moved between assemblers and customers than its sequential counterpart. This is due to the higher relevance of the transportation of components between manufacturers and assemblers in the total costs, as well as the larger number of assemblers present in the joint solution, which allow the network to supply customers from nearer assemblers.

These results remark that the joint solution does not only optimize the assembly costs, but also draws a stronger focus on the links between assemblers and customers. This change provides an intangible improvement (in terms of the objective costs) corresponding to a more responsive supply chain in which assemblers can react to customer needs in a more efficient way.

The main improvement provided by the joint resolution of the problem comes from the reduction of the assembly costs, the minimization of idle times and the maximization of the utilization ratio (flow of the assembler over the total capacity of the assembler). Metrics on assembly utilization are given in Table 9. Table 9 provides the minimum (column min.), average (column av.), maximum (column max.) and standard deviation (column sd.) on the number of units produced per assembler, the idle time of each assembler per cycle as well as the number of stations per assembler with positive production.

As it can be observed, while the sequential solution uses less assemblers (see Table 7, the number of opened stations per assembler is larger, and the efficiency of the assembly line is smaller (larger idle times). Thus, solving first the SCN design problem, to later balance the assembly lines seems to saturate all the assemblers, causing acute inefficiencies in terms of productivity in the assembler. Conversely, solving both problems jointly uses a higher number of assemblers than the alternative, but it uses them more efficiently, reducing the average idle time in the open stations. Note that this result is consistent with the conclusions from Section 4 and further highlights that the supply

Table 9. For each resolution method (Sequential and Joint), different metrics regarding the assemblers are provided. The minimum (column min.), average (av.), maximum (max.) and standard deviation (sd.) of the quantity of units produced per assembler, the idle time of each assembler and the number of stations per assembler are provided.

	Units produced per assembler				Idle time (# stations)			
	min.	av.	max.	sd.	min.	av.	max.	sd.
Sequential	387	3630	3956	934	0 (10)	77 (40)	86 (44)	26 (9)
Joint	277	2810	3070	603	0 (3)	80 (23)	10 (26)	3 (5)

chain of products with a large number of small assembly operations may benefit from the proposed approach.

6. Conclusions and future lines of work

In this paper, we have studied the joint supply chain network design and assembly line balancing design problem. We have proposed a decomposition approach which allows us to solve larger instances to optimality than those previously solved in the literature. A computational experiment conducted on a wide range of instances derived from the instance set proposed in Otto, Otto and Scholl (2013) for the SALBP, allows us to show the applicability and limitations of the proposed method, which is able to optimally solve instances with up to 50 tasks within reduced running times. Furthermore, we have considered the influence of the different costs, namely the transportation and the assembly cost, on the final configuration of the supply chain network, as well as the structure of the assembly process. The results show that some of these characteristics greatly affect the possible gains of jointly considering both problems.

The ability of the method to decompose line balancing and network design decisions led us to conclude that the integrated network design and assembly line balancing resolution is possible within the currently available computing resources, and that a larger level of detail, such as the consideration of larger time periods and/or inventory decisions within the supply chain, could be integrated into the resolution framework (these changes would only modify the mixed-integer linear program which can be efficiently solved using a commercial software). Furthermore the decomposition approach is also applicable to other line balancing problems, such as U-shaped, two-sided or other general line balancing problems (GALBPs), since the decomposition assimilates the optimal solutions of the specific assembly line problem into a piecewise linear function which is then integrated in the calculation of the network design problem.

We would also like to mention that the decomposition is also applicable, after some changes, to other supply network design problems in which the line balancing decisions are integrated into the network design, such as the studies by Hamta et al. (2015); Yolmeh and Salehi (2015). This possibility further highlights the applicability and usefulness of the proposed method.

In addition to the applicability of the proposed method for the supply network design problem with assembly line balancing decisions, the results of the computational experiments also provide some insights into the SALBP, which we proceed to analyse.

While there are several methods available to solve SALBP formulations (both SALBP-1 and SALBP-2), these methods may have difficulties, in terms of computing time, for some specific combinations of processing times. Therefore, the required running time for the available exact methods may be too high if, as it is the case in this work, one needs to identify the set of all the efficient solutions in terms of both cycle time and number of stations. While we used the state-of-the-art solver for the SALBP-1 to obtain such a set by enumeration of the different cycle times, the problem we are tackling has a multi-objective nature (both cycle time and number of stations have to be optimized) and thus a special method to reach the frontier of efficient solutions may be desirable (a multiobjective optimization method). Whereas multi-objective assembly line balancing problems have already been considered in the literature, see the review of Battaïa and Dolgui (2013), the identification of the efficient set of solutions according to the classical objectives

of the SALBP (cycle time and number of stations) is an open problem which will require additional research.

Finally, the results of our experiments show that significant gains can be obtained by jointly considering supply chain network design with line balancing decisions. These changes can be substantial (up to 16.9%) for those instances in which the assembly costs are high when compared to the transportation costs. Nevertheless, an average gain of 2% was obtained among the 4,725 instances considered in this study. **Furthermore, the solutions provided by the joint method offer higher levels of responsiveness as the number of assemblers increases and the distance between assemblers and customers decreases. Hence, the proposed solution approach does not only provide a better solution in terms of cost, but also provides an intangible improvement over classical sequential methods.**

References

- Battaïa, O., Dolgui, A. 2013. “A taxonomy of line balancing problems and their solution approaches”. *International Journal of Production Economics* 142: 259–77.
- Bautista, J., Pereira, J. 2009. “A dynamic programming based heuristic for the assembly line balancing problem”. *European Journal of Operational Research* 194: 787–794.
- Beamon, B.M. 1998. “Supply chain design and analysis: Models and methods”. *International Journal of Production Economics* 55: 281–294.
- Cachon, G., Terwiesch, C. 2013. *Matching supply with demand. An introduction to Operations Management*, 3rd edition, McGraw Hill, New York.
- Che, Z. G., Che, Z. H., Hsu, T. A. 2009. “Cooperator selection and industry assignment in supply chain network with line balancing technology”. *Expert Systems with Applications* 36: 10381–10387.
- Chopra, S., Meindl, P. 2012. *Supply Chain Management. Strategy, Planning and Operation*, 5th edition, Pearson, Essex.
- Emde, S., Boysen, N. 2012. “Optimally routing and scheduling tow trains for the JIT-supply of mixed-model assembly lines”. *European Journal of Operational Research* 217: 287–299.
- Faccio, M., Gamberi, M., Pilati, F., Bortolini, M. 2015. “Packaging strategy definition for sales kits within an assembly system”. *International Journal of Production Research* 53 (11): 3288–3305.
- Farahani, R. Z., Rezapour, S., Drezner, T., Fallah, S. 2014. “Competitive supply chain network design: An overview of classifications, models, solution techniques and applications”. *Omega* 45: 92–118.
- Farahani, R. Z., Rashidi Bajgan, H., Fahimnia, B., Kaviani, M. 2015. “Location-inventory problem in supply chains: a modelling review”. *International Journal of Production Research* 53: 3769–3788.
- Garey, M., Johnson D. 1979 *Computers and intractability: a guide to the theory of NP-completeness* New York; Freeman.
- Goetschalckx, M. Vidal, C.J. Dogan, K. 2002. “Modeling and design of global logistics systems: A review of integrated strategic and tactical models and design algorithms”. *European Journal of Operational Research* 143: 1–18.
- Hall, N. G., Potts, C. N. 2003. “Supply chain scheduling: Batching and delivery”. *Operations Research* 51, 566–584.
- Hamta, N., Akbarpour Shirazi, M., Fatemi Ghomi, S.M.T., Behdad, S. 2015. “Supply chain network optimization considering assembly line balancing and demand uncertainty”. *International Journal of Production Research* 53: 2970–2994.
- Klibi, W., Martel, A., Guitouni, A. 2010. “The design of robust value-creating supply chain networks: a critical review”. *European Journal of Operational Research* 203: 283–293.
- Limère, V., Van Landeghem, H., Goetschalckx, M., Aghezzaf, E-H., McGinnis, L. F. 2012. “Optimising part feeding in the automotive assembly industry: deciding between kitting and line stocking”. *International Journal of Production Research* 50 (15): 4046–4060.
- Meixell, M. J. Gargeya, V. B. 2005. “Global supply chain design: A literature review and critique”. *Transportation Research Part E: Logistics and Transportation Review* 41: 531–550.
- Melo, M. T., Nickel, S., Saldanha-Da-Gama, F. 2009. “Facility location and supply chain management A review”. *European Journal of Operational Research* 196: 401–412.
- Min, H., Zhou, G. 2002. “Supply chain modeling: past, present and future”. *Computers & Industrial Engineering* 43: 231–249.

- Morrison, D.R., Sewell, E.C., Jacobson, S.H. 2014. "An application of the branch, bound and remember algorithm to a new simple assembly line balancing dataset". *European Journal of Operational Research* 236: 403–409.
- Mula, J., Peidro, D., Díaz-Madroñero, M., Vicens, E. 2010. "Mathematical programming models for supply chain production and transport planning". *European Journal of Operational Research* 204: 377–390.
- Otto, A., Otto, C., Scholl, A. 2013. "Systematic data generation and test design for solution algorithms on the example of SALBPGen for assembly line balancing". *European Journal of Operational Research* 228: 33–45.
- Paksoy, T., Özceylan, E., Gökçen H. 2012. "Supply chain optimisation with assembly line balancing". *International Journal of Production Research* 50: 3115–3136.
- Paksoy, T. and Özceylan, E. 2012. "Supply chain optimisation with U-type assembly line balancing". *International Journal of Production Research* 50: 5085–5105.
- Pape, T. 2015. "Heuristics and lower bounds for the simple assembly line balancing problem type 1: Overview, computational tests and improvements". *European Journal of Operational Research* 240: 32–42.
- Pereira, J. 2015. "Empirical evaluation of lower bounding methods for the simple assembly line balancing problem". *International Journal of Production Research* 53: 3327–3340.
- Pishvaei, M. S., Farahani, R. Z., Dullaert, W. 2010. "A memetic algorithm for bi-objective integrated forward/reverse logistics network design". *Computers & Operations Research* 37: 1100–1112.
- Sawik, T. 2009. "Coordinated supply chain scheduling". *International Journal of Production Economics* 120: 437–451.
- Scholl, A., Becker, C. 2006. "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing". *European Journal of Operational Research* 168: 666–693.
- Sewell, E.C., Jacobson, S.H. 2012. "A branch, bound, and remember algorithm for the simple assembly line balancing problem". *INFORMS Journal on Computing* 24: 433–442.
- Srivastava, S. K. 2007. "Green supplychain management: a stateoftheart literature review". *International journal of management reviews* 9: 53–80.
- Sternatz, J. 2015 "The joint line balancing and material supply problem". *International Journal of Production Economics* 159: 304–318.
- Vilà, M., Pereira, J. 2013. "An enumeration procedure for the assembly line balancing problem based on branching by non-decreasing idle time". *European Journal of Operational Research* 229: 106–13.
- Yolmeh, A., Salehi, N. 2015. "An outer approximation scheme for an integration of supply chain network designing and assembly line balancing under uncertainty". *Computers & Industrial Engineering* 83: 297–306.