Les Cahiers du GERAD ISSN: 0711–2440

Efficient Formulations and a Branch-and-Cut Algorithm for a Production-Routing Problem

M. Ruokokoski, O. Solyali
J.-F. Cordeau, R. Jans,
H. Süral

G–2010–66

November 2010

Efficient Formulations and a Branch-and-Cut Algorithm for a Production-Routing Problem

Mirko Ruokokoski

Aalto University School of Science and Technology
Systems Analysis Laboratory
P.O. Box 11100 00076 Aalto, Finland
mirko.ruokokoski@tkk.fi

Oğuz Solyali

Business Administration Program,
Middle East Technical University, Northern Cyprus Campus
Kalkanli-Guzelyurt, Mersin 10, Turkey
solyali@metu.edu.tr

Jean-François Cordeau*
Raf Jans

*GERAD & CIRRELT**
HEC Montréal
3000 chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada, H3T 2A7
{raf.jans; jean-francois.cordeau}@hec.ca

Haldun Süral

Industrial Engineering Department
Middle East Technical University
06531 Ankara, Turkey
sural@ie.metu.edu.tr

November 2010

Les Cahiers du GERAD

G–2010–66

Abstract

The production-routing problem can be seen as a combination of two well known combinatorial optimization problems: the lotsizing and the vehicle routing problems. The variant considered in this paper consists in designing a production schedule for an uncapacitated plant, replenishment schedules for multiple customers, and a set of routes for a single uncapacitated vehicle starting and ending at the plant. The aim of the problem is to fulfill the demand of the customers over a finite horizon such that the total cost of distribution, setups, and inventories is minimized. This paper introduces a basic mixed integer linear programming formulation and several strong reformulations of the problem. Two families of valid inequalities, 2-matching and generalized comb inequalities, are introduced to strengthen these formulations, and they are used within a branch-and-cut algorithm. Computational results on a large set of randomly generated instances are presented. Instances with up to 40 customers and 15 time periods or with 80 customers and 8 periods have been solved to optimality within a two-hour limit. The tests clearly indicate the effectiveness of the new formulations and of the valid inequalities. In addition, an uncoordinated approach is considered to demonstrate the benefits of the simultaneous optimization of production and distribution planning. The total cost increases on average by 47% when employing such an uncoordinated approach. Finally, a heuristic algorithm, based on defining an a priori tour for the vehicle routing part, is investigated. The heuristic algorithm shows an excellent performance. The average CPU time is less than 1% of the CPU time for the optimal solution, whereas the average cost increase is only 0.33%.

Key Words: Vehicle routing, production planning, branch-and-cut algorithm, integer programming, reformulation techniques.

Résumé

Le problème de production et tournées peut être vu comme la combinaison de deux problèmes d'optimisation combinatoire bien connus: le problème de lotissement et le problème de tournées de véhicules. La variante considérée dans cet article consiste à concevoir un plan de production pour une usine à capacité infinie, un plan de réapprovisionnement des clients, et un ensemble de tournées de livraison pour un véhicule unique basé à l'usine. L'objectif du problème est de répondre à toute la demande des clients durant un horizon de planification fini de manière à minimiser le coût total des stocks, de la distribution et de mises en course. L'article présente une formulation linéaire en variables mixtes ainsi que plusieurs reformulations. Deux familles d'inégalités valides sont aussi proposées pour renforcer ces formulations et sont utilisées à l'intérieur d'un algorithme de séparation et coupes. Des résultats sont présentés sur un grand ensemble d'instances générées aléatoirement. Des instances avec 40 clients et un horizon de 15 périodes ou avec 80 clients et 5 périodes peuvent être résolues en deux heures de calcul. Une approche sans coordination est également présentée afin de démontrer les bénéfices de l'optimisation intégrée de la production et de la distribution. Finalement, un algorithme heuristique est étudié. Celui-ci exige seulement 1% du temps nécessaire à la résolution exacte du problème et fournit des solutions avec un coût ne dépassant en moyenne que de 0.33% le coût de la solution optimale.

Mots clés : tournées de véhicules, planification de la production, séparation et coupes, programmation en nombres entiers, techniques de reformulation.

Acknowledgments: This work was supported by the Research Foundation of Helsinki University of Technology as well as by the Natural Sciences and Engineering Research Council of Canada under grants 227837-09 and 342182-09. These supports are gratefully acknowledged.

1 Introduction

A supply chain is a network of facilities that procure raw materials, transform them into finished products, and deliver the products to customers through a distribution system to meet the demand of the customers. Supply chain management consequently deals with a collection of interrelated subproblems such as raw material procurement, production planning, inventory management, and distribution routing. These subproblems have been extensively researched but mostly separately dealt with in both industrial applications and the literature. In the last three decades several studies have shown that coordinating these subproblems can lead to significant cost savings in the supply chain. In this paper we consider a joint production and distribution planning problem. This is a relevant problem in practice, as witnessed by the examples of many companies that considered joint production and distribution planning. For example, Blumenfeld et al. (1987) investigated an optimization model that synchronizes scheduling of production and distribution at the Delco electronics division of General Motors, which resulted in a 26% reduction in logistics costs. Similar results have been reported for other companies such as Xerox (Stenross and Sweet, 1991), Hewlett Packard (Lee and Billington, 1995), Kellog Company (Brown et al., 2001), and Frito-Lay (Çetinkaya et al., 2009).

More specifically, we investigate the Production-Routing Problem (PRP) which considers the simultaneous coordination of production planning and distribution routing. These two subproblems are known in the literature as the lotsizing problem and the inventory-routing problem, respectively. The PRP can also be classified as a combination of a two-level lotsizing problem and a vehicle routing problem. The PRP comprises a plant, producing a set of products using resources. If there is production in a specific time period, then setup and production costs are incurred. Production costs are dependent on the production volume, whereas the setup cost is independent of the realized volume. Resources in a production system can include, e.g., manpower, equipment, machines, budget, etc. The finished products are delivered to customers to meet the time-varying demand. Deliveries can be made to a customer before the actual (external) demand realizes, but backlogging is not allowed. Any excess production, both at the plant and at a customer, is carried over as inventory to the next period and inventory holding costs are incurred. Therefore, the demand in a specific time period at a customer site can be met from deliveries in the period and from the inventories left over from the previous period at the customer site. Transportation from the plant to customers with a multi-stop routing policy is carried out by a fleet of vehicles. If a vehicle leaves the plant to make deliveries to a set of customers, then a routing cost is incurred that is proportional to the traveled distance.

The PRP consists in designing a production schedule at the plant, replenishment schedules for multiple customers, and a set of vehicle routes starting and ending at the plant. The planning is done over a finite discrete time horizon in a coordinated manner. The aim of the problem is to meet the demand of the customers in such a way that the total cost of distribution, setups, and inventories is minimized subject to the constraints of the problem.

The PRP is NP-hard as it contains the Traveling Salesman Problem (TSP) as a special case. As a result, most of the previous studies on the PRP have focused on heuristic solution approaches. To the best of our knowledge, only a few studies, such as Fumero and Vercellis (1999), Archetti et al. (2009), and Bard and Nananukul (2010), consider obtaining lower bounds. All of these studies use a weak representation of the problem which limits their ability to obtain optimal solutions.

Our focus in this paper is on exact methods and strong formulations. We study the PRP in which the following assumptions hold. The plant manufactures a single product using resources that are available in unlimited quantity in each time period. Replenishments from the plant to the customers are performed by a single uncapacitated vehicle. Routing costs are symmetrical. We assume that the lead times are zero. The demand at the customer sites is considered to be deterministically known in each time period. Shortages are not permitted. We assume that initial inventories are equal to zero and that all the costs related to production, inventory holding as well as routing are time-invariant. Since we assume that production costs are time-invariant, the total production cost over the whole horizon is constant. Therefore, the production cost is omitted from the problem. These restrictions do not limit the flexibility of our formulations and solution approach. Although the problem being considered is simple, it is important to analyze it as it will form the building block for extensions such as the multi-vehicle case or the case with capacity constraints for the vehicles and for the plant. Since our solution approach is based on strong formulations and cutting planes, it can also be used to solve extended problems that contain the basic PRP as a substructure.

We make several contributions in this paper. First, we introduce strong formulations for the PRP. Second, we present a branch-and-cut algorithm to solve these formulations. We introduce two families of valid inequalities adapted from the literature, 2-matching and generalized comb inequalities, to strengthen these formulations. Third, we present a new heuristic separation algorithm for the generalized comb inequalities. Fourth, we adapt a heuristic algorithm presented by Solyalı and Süral (2008) to find high quality integer feasible solutions for the PRP. Fifth, we report extensive computational experiments on a large set of instances showing that strong formulations both increase the lower bound and decrease the CPU time, and that the simultaneous optimization of production and distribution reduces costs significantly.

The content of the paper is as follows. In the next section we give a literature review on the previous studies related to the PRP. In Section 3 we introduce the notation, the basic formulation, several strong reformulations of the problem, and an uncoordinated approach. In Section 4 we introduce two families of valid inequalities adapted from the literature to strengthen the formulations. These valid inequalities are added to the model during the branch-and-cut algorithm which is described in Section 5. The a priori tour heuristic algorithm adapted from an inventory-routing context is introduced in Section 6. Computational experiments are reported in Section 7 and conclusions follow in Section 8.

2 Literature Review

As mentioned earlier, the PRP can be seen as a combination of the Lotsizing Problem (LSP) and the Inventory-Routing Problem (IRP). Both the IRP and LSP have been extensively studied in isolation. For more information about the IRP and LSP, the reader is referred to the reviews of Andersson et al. (2010) and Jans and Degraeve (2008), respectively. In the PRP, production (i.e., lotsizing) decisions should be given at the plant in addition to the classical IRP such as routing decisions and replenishment decisions at the retailers. Although there is a vast amount of literature on the IRP (see, e.g., Andersson et al. 2010), the literature on the PRP is rather limited but growing rapidly as will be seen in this section. In the following we give a review on the PRP, in which inventory management at customer and plant sites as well as production planning and distribution routing are involved.

We will review the literature on the PRP from three different angles. First, we discuss the various components which can be taken into account in the objective function. Second, we give an overview of the many extensions that have been proposed. Finally, we review the various solution approaches that have been used. This discussion will allow us to clearly show the value of our various contributions.

As the PRP is a combination of the classical lotsizing and vehicle routing problems, the traditional cost components related to these two substructures are typically found as well in the PRP. Almost all the works have inventory holding and set up costs at the plant level. The only study that does not consider set up costs at the plant level is Lei et al. (2006). Also the inventory holding cost at the customer level is a standard cost element. However, Boudia et al. (2007, 2008), and Boudia and Prins (2009) do not consider this cost as they assume that it is paid by the customer. Variable production costs at the plant level are usually not explicitly taken into account. In most situations, these costs are assumed to be time-invariant and hence the total production cost over the whole time horizon is a fixed amount and can be left out. However, it is straightforward to incorporate these costs in the models (Bertazzi et al. 2005, Lei et al. 2006, Archetti et al. 2009, Shiguemoto and Armentano 2010). On the routing side, fixed transportation costs can be incurred if a vehicle is used in a specific time period (Chandra 1993, Chandra and Fisher 1994, Fumero and Vercellis 1999). Alternatively, Bertazzi et al. (2005) and Shiguemoto and Armentano (2010) assume a fixed cost if a vehicle is used at least once over the whole time horizon. Variable transportation costs are mostly solely distance-based, except in Fumero and Vercellis (1999) where these costs are assumed to be based on the distance and the load. Lei et al. (2006) include both types of variable transportation costs in their model. All studies consider the asymmetric version of the problem, that is, the distance between any pair of locations is not necessarily the same in the two directions. Finally, fixed delivery costs per customer are only considered by Bard and Nananukul (2009b) in a relaxation of the full model.

Many extensions to the basic PRP have been proposed. A natural extension on the production side is to include multiple items (Chandra 1993, Chandra and Fisher 1994, Fumero and Vercellis 1999, Shiguemoto and Armentano 2010). Many authors also consider limited production capacity (Chandra and Fisher 1994, Fumero and Vercellis 1999, Lei et al. 2006, Bard and Nananukul 2009a; 2009b, and 2010, Boudia et al. 2007, 2008, and Boudia and Prins 2009). In some cases, there is also a capacity limit on the inventory level (Lei et al. 2006, Bertazzi et al. 2005, Archetti et al. 2009, Bard and Nananukul 2009a; 2009b, and 2010, Boudia et al. 2007, 2008, and Boudia and Prins 2009, Shiguemoto and Armentano 2010). Bertazzi et al. (2005) consider an order-up-to level policy imposing that if a delivery is made to a specific customer, the customer's inventory has to be raised to a pre-specified level. Archetti et al. (2009) compare this policy to a policy without such restriction. All works also consider only a single plant, except Lei et al. (2006). In none of the works have we found the option of backlogging the demand. On the routing part, most studies consider a fleet of capacitated vehicles. Only Lei et al. (2006) incorporate a fleet of heterogeneous vehicles. A further extension is to allow split deliveries (Chandra 1993, Chandra and Fisher 1994, Fumero and Vercellis 1999).

Since the PRP and its extensions are very hard to solve, most of the research has focused on heuristic algorithms. Many studies propose a sequential heuristic in which the problem is first solved using an uncoordinated approach. The lotsizing and routing subproblems are solved sequentially (optimally or heuristically), and the result of the first problem is an input for the second problem. Next, this solution is usually improved further to better coordinate the production and routing decisions. In some works (Chandra 1993, Chandra and Fisher 1994, Boudia et al. 2008), the production problem is first solved, and next the routing part is solved. Sometimes the production problem is further improved in a third step (Bertazzi et al., 2005). Other works (Bertazzi et al. 2005, Archetti et al. 2009) first heuristically solve the distribution problem, and next the production subproblem. Bertazzi et al. (2005) prove that for a specific choice of initial demand values, both sequences lead to the same result if the subproblems are solved to optimality. Lei et al. (2006) propose a different two-phase heuristic, where first the problem is optimally solved allowing only direct shipments and next this solution is heuristically improved by allowing combined shipments. Bard and Nananukul (2009b) also first solve a relaxation of the problem where only direct shipment costs (fixed delivery cost per customer as well as a volume and distance based variable cost) are considered and propose a tabu search algorithm to construct good routes. Various other meta-heuristics have also been proposed for extensions of the PRP such as a tabu search algorithm by Shiguemoto and Armentano (2010), a memetic algorithm by Boudia and Prins (2009), and a GRASP by Boudia et al. (2007).

To the best of our knowledge, only two studies propose exact algorithms. Bard and Nananukul (2010) propose a branch-and-price algorithm in which they consider the lotsizing and inventory constraints as the complicating constraints which are kept in the master. The subproblem decomposes into separate routing problems per time period. A heuristic implementation of the branch-and-price algorithm is also investigated by Bard and Nananukul (2009a). A branch-and-cut algorithm has been developed by Archetti et al. (2009). Some valid inequalities are added a priori to the formulation and subtour elimination constraints are dynamically added. Using exact algorithms, only small instances can be solved to optimality. The application of exact algorithms is limited to 6 time periods and 10 customers (Bard and Nananukul, 2010) and 6 time periods and 19 customers (Archetti et al., 2009). To solve larger problems, the authors in both papers resort to heuristics.

Some studies (Bard and Nananukul 2009b and Boudia et al. 2007) note that the LP lower bound from the standard formulation gives very weak values and leads to integrality gaps of 50% and more. All studies in the literature use this weak standard formulation. Apart from the two works using exact algorithms, only a few studies investigate stronger lower bounds. Fumero and Vercellis (1999) obtain a lower bound by applying Lagrangean relaxation. They relax the demand balance constraint at the plant and the vehicle capacity constraints. A specific heuristic is devised to also obtain feasible solutions. To solve the uncoordinated approach, they use a two-phase heuristic, in which the lotsizing part is solved optimally and the remaining distribution problem is solved via a Lagrangean heuristic. Bard and Nananukul (2009b) propose a lower bound based on solving a relaxation of the model in which only direct shipment costs are considered.

Fumero and Vercellis (1999) note that the comparison of the coordinated and uncoordinated approaches cannot be considered accurate as both are solved using heuristic methods (as also done by Chandra 1993 and by Chandra and Fisher 1994). Our paper is the first study to compare optimal solutions for both the coordinated and the uncoordinated approaches.

3 Formulations

We begin this section by presenting the general notation used in the paper. Additional notation will be defined as needed in subsequent subsections. The first formulation that we present is a basic model of the PRP. This formulation, called WEAK, will serve as a comparison basis for all other formulations. As indicated before, it is used in all other studies discussing the PRP. In the subsequent subsections we develop strong reformulations, i.e., formulations for which the optimal value of the linear programming relaxation is closer to the optimal integer solution value than that of the original formulation.

For the uncapacitated lotsizing problem, two reformulations have been introduced: the *shortest path* (Eppen and Martin, 1987) and the *facility location* formulations (Krarup and Bilde, 1977). These reformulations can be applied to a wide range of lotsizing problems (Pochet and Wolsey, 2006). Using the same ideas, we obtain two new reformulations of the PRP. We call them the *Four Index Shortest Path* (FISP) and the *Four Index Facility Location* (FIFL) formulations, respectively, as four indices are involved in the redefined variables.

The FISP and FIFL formulations have one disadvantage, namely, the high number of variables due to four indices. We thus present two formulations that involve only three-index variables. This can be done by reformulating the plant and customer lotsizing subproblems separately. However, we cannot do this directly since the demand at the plant is not known a priori but it is actually equal to the deliveries to the customers, which are decision variables. To counter this problem, we use the echelon stock reformulation.

The *echelon concept*, introduced by Clark and Scarf (1960), has often been used to decompose multi-level lotsizing problems into independent single level lotsizing subproblems (see, e.g., Pochet and Wolsey 2006, chapter 13). Note that the PRP can be seen as a two-level problem where the plant is the first level, and the customers are the second level. Therefore, we first use the echelon concept to decompose the problem into independent single-item lotsizing subproblems for each customer and the plant. We next use the previously mentioned reformulation ideas to obtain two new reformulations, called the *Echelon Stock Facility Location* (ESFL) and the *Echelon Stock Shortest Path* (ESSP) formulations, respectively.

These four formulations are used for the first time for the PRP. In the context of the one-warehouse multi-retailer problem, however, similar formulations have already been used (see, e.g., Federgruen and Tzur 1999 for the echelon stock reformulation, Levi et al. 2008 for the four index facility location reformulation, and Solyalı and Süral 2009 for the four index shortest path formulation).

Another way to obtain reformulations is to add valid inequalities (cutting planes) a priori to the basic formulation in the original variables. In our fifth reformulation we consider adding the so-called (l, S) inequalities. These valid inequalities are known to describe the convex hull of feasible solutions for the uncapacitated single-item lotsizing problem (see Barany et al. 1984). However, it is not practical to add all (l, S) inequalities since their cardinality is exponential with respect to the number of time periods. If an uncapacitated single-item lotsizing problem has *Wagner-Whitin Costs* (WWC), i.e., the sum of production and holding costs in a period is greater than or equal to the production cost in the next period, then only a small subset of (l, S) inequalities is needed to describe the convex hull of feasible solutions. Those inequalities are called (l, S, WW) inequalities (see, e.g., Pochet and Wolsey 2006, page 225).

In our fifth reformulation, we first use the echelon stock concept to decompose the basic problem into independent single-item lotsizing problems for each customer and the plant. Then, we add (l, S, WW) inequalities to each subproblem. We refer to the resulting formulation as the *Echelon Stock Wagner-Whitin cost* (ESWW) formulation. Since the (l, S, WW) inequalities are a subset of the general (l, S) inequalities, they are valid even if the WWC assumption does not hold. In Section 3.8 we explain in detail when the WWC assumption holds in the PRP problem.

Since we want to compare optimal solutions for the coordinated and the uncoordinated approaches, we also present the uncoordinated approach for the PRP in the last part of this section. As the uncoordinated approach has been studied widely, we only give a short description of it.

3.1 Notation

The PRP can be defined on a complete undirected graph $G = (N, E)$ where $N = N_c \cup \{0\}$ is the set of nodes, $N_c = \{1, 2, \ldots, n\}$ is the subset of customer nodes, node 0 represents the plant, and $E = \{(i, j) : i, j \in N, i < j\}$ is the set of edges. We define the set $T = \{1, \ldots, m\}$ of time periods. With each edge $(i, j) \in E$ is associated a routing cost c_{ij}. For each node $i \in N_c$ and time period $t \in T$, let d_{it} denote the demand of customer i in period t. We denote by h_i the holding cost at node $i \in N$ and by s the setup cost at the plant.

Several types of decision variables are defined. We let I_{it} be the inventory level at node $i \in N$ at the end of period $t \in T$. Note that $I_{i0} = 0 \ \forall i \in N$. We let r_{it} represent the amount delivered to node $i \in N_c$ in period $t \in T$ and p_t the production level in period $t \in T$. For each edge $(i, j) \in E$ with $i \neq 0$, we let x_{ijt} be a binary variable taking value 1 iff the vehicle travels along edge (i, j) in period $t \in T$. For $i = 0$, this variable is ternary and takes value 2 iff the vehicle visits only customer j in period $t \in T$. We let y_t be a binary variable taking value 1 iff there is a setup at the plant in period $t \in T$ and z_{it} a binary variable taking value 1 iff node $i \in N$ is visited in period $t \in T$.

We also define $\delta(S) = \{(i, j) \in E : i \in S, j \notin S \text{ or } i \notin S, j \in S\}$ as the set of edges incident to a node set S and $x_t(S) = \sum_{i,j \in S} x_{ijt}$ as the total flow on the edges in set S in period t to simplify the notation. We denote by $d_{0t} = \sum_{i \in N_c} d_{it}$ the aggregated demand of all customers in period $t \in T$, and by $D_{itu} = \sum_{q=t}^{u} d_{iq}$ the sum of the demand at node $i \in N$ from period t to u.

3.2 Basic Formulation

The basic formulation of the PRP is as follows:
Minimize

$$\sum_{t \in T} \left(s \times y_t + \sum_{i \in N} h_i I_{it} + \sum_{(i,j) \in E} c_{ij} x_{ijt} \right) \tag{1}$$

subject to

$$\begin{aligned}
& p_t \leq D_{0tm} y_t, && \forall t \in T, && (2)\\
& I_{0,t-1} + p_t = \sum_{i \in N_c} r_{it} + I_{0t}, && \forall t \in T, && (3)\\
& I_{i,t-1} + r_{it} = d_{it} + I_{it}, && \forall i \in N_c, \forall t \in T, && (4)\\
& r_{it} \leq D_{itm} z_{it}, && \forall i \in N_c, \forall t \in T, && (5)\\
& x_t(\delta(i)) = 2 z_{it}, && \forall i \in N, \forall t \in T, && (6)\\
& z_{0t} \geq z_{it}, && \forall i \in N_c, \forall t \in T, && (7)\\
& x_t(\delta(S)) \geq 2 z_{it}, && \forall S \subseteq N_c, |S| \geq 2, \forall i \in S, \forall t \in T, && (8)\\
& I_{it} \geq 0, && \forall i \in N, \forall t \in T, && (9)\\
& r_{it} \geq 0, && \forall i \in N_c, \forall t \in T, && (10)\\
& p_t \geq 0, && \forall t \in T, && (11)\\
& x_{ijt} \in \{0, 1\}, && \forall (i, j) \in E : i \neq 0, \forall t \in T, && (12)\\
& x_{0jt} \in \{0, 1, 2\}, && \forall j \in N_c, \forall t \in T, && (13)\\
& z_{it} \in \{0, 1\}, && \forall i \in N, \forall t \in T, && (14)\\
& y_t \in \{0, 1\}, && \forall t \in T. && (15)
\end{aligned}$$

The objective function (1) minimizes the total cost of setups, inventories, and distribution. If there is production in a specific time period, then a setup cost is incurred at the plant as ensured by constraints

(2). Constraints (3) and (4) are the inventory balance constraints at the plant and customer locations, respectively. If there is delivery to a node, then it must be visited in that time period as forced by constraints (5). Constraints (6) require the number of edges incident to a node to be either 2 if node i is visited or 0 otherwise. Constraints (7) ensure that if a customer node is visited, then the depot is also visited since the vehicle starts and ends its route at the plant. The Generalized Subtour Elimination Constraints (GSECs) (8) eliminate subtours; there must be at least two edges between S and its complement whenever node $i \in S$ belongs to the solution and the plant does not belong to S. Inventory, delivery, and production level variables are nonnegative. All the setup and routing variables are binary, except the routing variables associated with edges that are incident to the plant. Those variables are ternary to allow single customer trips.

The first part of the model, i.e., constraints (2) – (5), represents a one-warehouse multi-retailer lotsizing problem (see, e.g., Solyalı and Süral 2009). The second (routing) part of the model, i.e., constraints (6) – (8), is adapted from the orienteering problem context (see, e.g., Fischetti et al. 1998). The disadvantage of this part is the presence of GSECs, whose size grows exponentially with the number of customers. This disadvantage is, however, mitigated by the fact that not all GSECs must be put into the formulation at the beginning, but they can be generated as needed. There are two main differences between our routing formulation and the PRP formulations used in the literature. These formulations consider the asymmetric version of the problem, i.e., the distance between each pair of locations is not necessarily the same in the two directions. However, all the studies which use a distance-based variable transportation cost actually assume symmetric costs in their computational experiments. The advantage of the symmetric formulation is that it contains fewer routing variables, but we need ternary variables to represent single customer trips. The other difference is related to subtour elimination. In Fumero and Vercellis (1999), Lei et al. (2006), and Shiguemoto and Armentano (2010) subtours are avoided by commodity flow conservation constraints, the number of which grows polynomially. As expected, the LP relaxation of this formulation is inferior to the relaxation of the formulations with GSECs. In Chandra and Fisher (1994) capacity constraints are used to avoid subtours, the size of which grows exponentially. In this case, the minimum number of vehicles needed to serve any set of customers, i.e., the right hand side of the constraint, cannot be rounded up as the quantity of the product delivered to customers in a period is not known in advance. This makes its relaxation weaker. Boudia et al. (2007, 2008) and Boudia and Prins (2009) eliminate subtours using the traditional subtour elimination constraints, while Archetti et al. (2009) use very similar GSECs as we do.

3.3 Four Index Facility Location Formulation (FIFL)

To describe the FIFL formulation, we introduce an additional decision variable f_{ituq} representing the amount produced in period t that is delivered to customer i in period u to satisfy demand in period q. The FIFL formulation of the PRP is as follows:
Minimize (1)
subject to (6) – (8), (12) – (15), and

$$\sum_{u=1}^{t}\sum_{q=u}^{t} f_{iuqt} = d_{it}, \qquad \forall t \in T, \forall i \in N_c, \tag{16}$$

$$\sum_{q=u}^{t} f_{iuqt} \leq d_{it} y_u, \qquad \forall i \in N_c, 1 \leq u \leq t \leq m, \tag{17}$$

$$\sum_{u=1}^{q} f_{iuqt} \leq d_{it} z_{iq}, \qquad \forall i \in N_c, 1 \leq q \leq t \leq m, \tag{18}$$

$$I_{0,t-1} + \sum_{i \in N_c}\sum_{q=t}^{m}\sum_{u=q}^{m} f_{itqu} = \sum_{i \in N_c}\sum_{q=1}^{t}\sum_{u=t}^{m} f_{iqtu} + I_{0t}, \qquad \forall t \in T, \tag{19}$$

$$I_{i,t-1} + \sum_{q=1}^{t}\sum_{u=t}^{m} f_{iqtu} = d_{it} + I_{it}, \qquad \forall t \in T, \forall i \in N_c, \tag{20}$$

$$f_{ituq} \geq 0, \qquad \forall i \in N_c, 1 \leq t \leq u \leq q \leq m. \tag{21}$$

Constraints (16) ensure that the total amount produced at the plant and delivered to i from period 1 through t is equal to the demand of i in t. Constraints (17) and (18) are used in place of (2) and (5), respectively. Constraints (19) and (20) are inventory balance constraints at the plant and customer sites, respectively. Notice that constraints (19) and (20) can be eliminated by rewriting the objective function in terms of the new f_{ituq}. Constraints (21) are nonnegativity constraints. The link with the variables of the basic formulation is as follows:

$$p_t = \sum_{i \in N_c} \sum_{q=t}^{m} \sum_{u=q}^{m} f_{itqu}, \quad r_{it} = \sum_{q=1}^{t} \sum_{u=t}^{m} f_{iqtu}.$$

3.4 Four Index Shortest Path Formulation (FISP)

Following Eppen and Martin (1987), we define g_{ituq} as the fraction of the demand at node $i \in N_c$ from period u through q that is produced in period t and delivered to node i in period u. We let a_{itu} be equal to 1 if $D_{itu} > 0$ and 0 otherwise. The FISP formulation of the PRP is as follows:
Minimize (1)
subject to (6) – (8), (12) – (15), and

$$\sum_{q=1}^{m} g_{i11q} = 1, \qquad \forall i \in N_c, \tag{22}$$

$$\sum_{q=1}^{t-1} \sum_{u=q}^{t-1} g_{iqu,t-1} = \sum_{q=1}^{t} \sum_{u=t}^{m} g_{iqtu}, \qquad \forall i \in N_c, 2 \le t \le m, \tag{23}$$

$$\sum_{q=1}^{t} \sum_{u=t}^{m} a_{itu} g_{iqtu} \le z_{it}, \qquad \forall i \in N_c, \forall t \in T, \tag{24}$$

$$\sum_{l=u}^{t} \sum_{q=t}^{m} a_{ilq} g_{iulq} \le y_u, \qquad \forall i \in N_c, 1 \le u \le t \le m, \tag{25}$$

$$I_{0,t-1} + \sum_{i \in N_c} \sum_{u=t}^{m} \sum_{q=u}^{m} D_{iuq} g_{ituq} = \sum_{i \in N_c} \sum_{u=1}^{t} \sum_{q=t}^{m} D_{itq} g_{iutq} + I_{0t}, \qquad \forall t \in T, \tag{26}$$

$$I_{i,t-1} + \sum_{u=1}^{t} \sum_{q=t}^{m} D_{itq} g_{iutq} = d_{it} + I_{it}, \qquad \forall i \in N_c, \forall t \in T, \tag{27}$$

$$g_{ituq} \ge 0, \qquad \forall i \in N_c, 1 \le t \le u \le q \le m. \tag{28}$$

Constraints (22) – (23) are the conservation of flow equalities for the shortest path network. Figure 1 illustrates the shortest path network for customer i in a problem instance with three periods. Sending a unit flow through the network is equivalent to imposing that demand must be met within the horizon. For instance, if $g_{i113} = 1$, then production in period 1 satisfies the demand at node i from period 1 to 3 and the production is delivered to node i in period 1. Constraints (24) and (25) are used in place of (2) and (5), respectively. Note that if $D_{itu} = 0$, then (24) and (25) do not force setups as a_{itu} is equal to zero in this case. For any fixed $u = \hat{u}$ and $t = \hat{t}$ the left hand side of constraints (25) is the flow from the set $\{\hat{u} - 1, \ldots, \hat{t} - 1\}$ to the set $\{\hat{t}, \ldots, m\}$ in the shortest path network related to production in period $\hat{u}$. Due to the definition of the shortest path network, this flow is always less than or equal to one. Constraints (26) and (27) are the inventory balance constraints at the plant and customer sites, respectively. Constraints (28) are the nonnegativity constraints. The link with the variables of the basic formulation is as follows:

$$p_t = \sum_{i \in N_c} \sum_{u=t}^{m} \sum_{q=u}^{m} D_{iuq} g_{ituq}, \quad r_{it} = \sum_{u=1}^{t} \sum_{q=t}^{m} D_{itq} g_{iutq}.$$

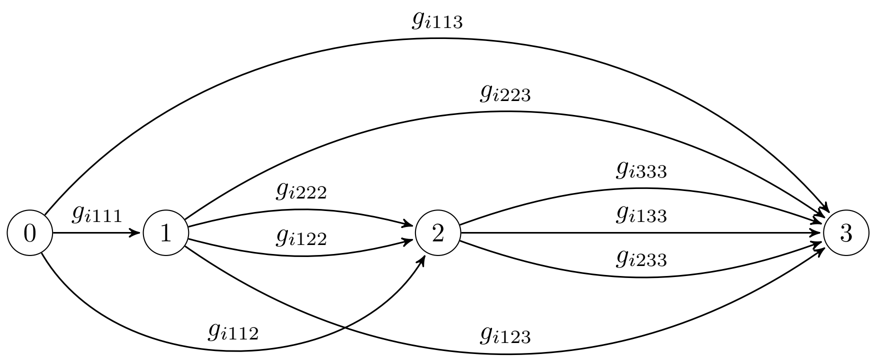

Figure 1: Shortest path network for customer i in a problem with 3 periods.

3.5 Echelon Stock Reformulation

Multi-level lotsizing problems can be decomposed into a set of independent single level lotsizing subproblems by applying the echelon stock concept in which each inventory variable of an item at a specific level is replaced by an echelon stock variable representing the amount of inventory of the item, not only at that level, but also at all subsequent levels where it is present (see, e.g., Pochet and Wolsey 2006, Section 13.1.2). Therefore, echelon stock variables for the PRP can be defined as follows:

$$
\begin{aligned}
e_{it} &= I_{it}, && \forall i \in N_c, \forall t \in T,\\
e_{0t} &= \sum_{i\in N} I_{it}, && \forall t \in T.
\end{aligned}
$$

These equations mean that for a customer an echelon stock variable is the same as an inventory variable whereas for the plant, an echelon stock variable states the total stock of the item within the whole system, i.e., both at the plant and customer level, at time t.

Replacing the regular inventory variables by the echelon stock variables results in the following formulation:
Minimize

$$\sum_{t\in T}\left(s \times y_t + h_0 e_{0t} + \sum_{i\in N_c}(h_i - h_0)e_{it} + \sum_{(i,j)\in E} c_{ij}x_{ijt}\right) \tag{29}$$

subject to (2), (5) – (8), (10) – (15), and

$$e_{0,t-1} + p_t = e_{0t} + d_{0t}, \qquad \forall t \in T, \tag{30}$$

$$e_{i,t-1} + r_{it} = e_{it} + d_{it}, \qquad \forall t \in T, \forall i \in N_c, \tag{31}$$

$$e_{0t} \geq \sum_{i\in N_c} e_{it}, \qquad \forall t \in T, \tag{32}$$

$$e_{it} \geq 0, \qquad \forall i \in N, \forall t \in T. \tag{33}$$

The objective function (29) is rewritten in terms of the echelon stock variables. Constraints (30) and (31) are the demand balance constraints at the plant and customer level, respectively. Note that now the plant is facing known demand, i.e., the aggregated demand of all the customers. Constraint (32) ensure that the echelon stock at the plant (i.e., the total stock in the system) is greater than or equal to the sum of the inventories at the customer sites. These constraints result from reformulating the original non-negativity constraint on the plant inventory level using the echelon stock variables via the following substitution:

$$I_{0t} = e_{0t} - \sum_{i\in N_c} I_{it} = e_{0t} - \sum_{i\in N_c} e_{it} \geq 0.$$

This reformulation will not give a better LP relaxation than the basic formulation. Therefore, we will not use this formulation in our computational experiments. The advantage is, however, that we now have independent single item uncapacitated lotsizing problems both at the plant and customer level as a substructure. In the next three sections, we will apply known reformulation techniques that give stronger bounds for the single item uncapacitated lotsizing substructures. Without using the echelon stock concept, this would not be possible as the demand would not be known at the plant level.

3.6 Echelon Stock Facility Location Formulation (ESFL)

We now can reformulate the independent uncapacitated single item lotsizing problems at both the plant and customer levels using the Facility Location Formulation. Let v_{0ut} be the amount produced at the plant in period u to satisfy the aggregated demand of all the customers in period t and v_{iut} the amount delivered to customer i in period u to satisfy the demand of customer i in period t. The ESFL formulation of the PRP is as follows:
Minimize (29)
subject to (6) – (8), (12) – (15), (32) – (33), and

$$\sum_{u=1}^{t} v_{iut} = d_{it}, \qquad \forall i \in N, \forall t \in T, \tag{34}$$

$$v_{0ut} \leq d_{0t} y_u, \qquad 1 \leq u \leq t \leq m, \tag{35}$$

$$v_{iut} \leq d_{it} z_{iu}, \qquad \forall i \in N_c, 1 \leq u \leq t \leq m, \tag{36}$$

$$e_{0,t-1} + \sum_{u=t}^{m} v_{0tu} = e_{0t} + d_{0t}, \qquad \forall t \in T, \tag{37}$$

$$e_{i,t-1} + \sum_{u=t}^{m} v_{itu} = e_{it} + d_{it}, \qquad \forall t \in T, \forall i \in N_c, \tag{38}$$

$$v_{iut} \geq 0, \qquad \forall i \in N, 1 \leq u \leq t \leq m. \tag{39}$$

Constraints (34) ensure that the demand in each time period and for each customer is satisfied. Constraints (35) stipulate that a setup is incurred if there is production in period u. If there is a delivery to a node, it must be visited in that period (36). Constraints (37) and (38) are inventory balance constraints at the plant and customer sites, respectively. Constraint (39) are nonnegativity constraints. The link with the variables of the basic formulation is as follows:

$$p_t = \sum_{u=t}^{m} v_{0tu}, \quad r_{it} = \sum_{u=t}^{m} v_{itu}.$$

3.7 Echelon Stock Shortest Path Formulation (ESSP)

Let w_{0tu} be the fraction of the aggregated demand of all customers from period t through u that is produced in period t and w_{itu} the fraction of the demand of customer i from period t through u that is delivered in period t.

Recall that constants a_{itu} and variables e_{it} are defined in Sections 3.4 and 3.6, respectively. The ESSP formulation of the PRP is as follows:
Minimize (29)

subject to (6) – (8), (12) – (15), (32) – (33), and

$$\sum_{t=1}^{m} w_{i1t} = 1, \qquad \forall i \in N, \tag{40}$$

$$\sum_{u=1}^{t-1} w_{iu,t-1} = \sum_{u=t}^{m} w_{itu}, \qquad \forall i \in N, 2 \leq t \leq m, \tag{41}$$

$$\sum_{u=t}^{m} a_{0tu} w_{0tu} \leq y_t, \qquad \forall t \in T, \tag{42}$$

$$\sum_{u=t}^{m} a_{itu} w_{itu} \leq z_{it}, \qquad \forall t \in T, \forall i \in N_c, \tag{43}$$

$$e_{0,t-1} + \sum_{u=t}^{m} D_{0tu} w_{0tu} = e_{0t} + d_{0t}, \qquad \forall t \in T, \tag{44}$$

$$e_{i,t-1} + \sum_{u=t}^{m} D_{itu} w_{itu} = e_{it} + d_{it}, \qquad \forall t \in T, \forall i \in N_c, \tag{45}$$

$$w_{itu} \geq 0, \qquad \forall i \in N, \forall 1 \leq t \leq u \leq m. \tag{46}$$

The flow conservation is ensured by constraints (40) – (41). Constraints (42) define the production setup variables y_t whereas constraints (43) define the delivery setup variables z_{it}. Note that if $D_{ilm} = 0$, then (42) – (43) do not force setups as a_{ilm} is zero. Constraints (44) – (45) express the demand satisfaction in each time period. Constraints (46) are nonnegativity constraints. The link with the variables of the basic formulation is as follows:

$$p_t = \sum_{u=t}^{m} D_{0tu} w_{0tu}, \quad r_{it} = \sum_{u=t}^{m} D_{itu} w_{itu}.$$

3.8 Echelon Stock Formulation with Wagner-Whitin Costs (ESWW)

This formulation is the same as the basic echelon stock reformulation in Section 3.5 with the following (l, S, WW) inequalities added at both plant and customer level:

$$e_{0,t-1} + \sum_{q=t}^{u} D_{0qu} y_q \geq D_{0tu}, \qquad 1 \leq t \leq u \leq m, \tag{47}$$

$$e_{i,t-1} + \sum_{q=t}^{u} D_{iqu} z_{iq} \geq D_{itu}, \qquad 1 \leq t \leq u \leq m, \forall i \in N_c. \tag{48}$$

Constraints (47) indicate that if there is no production during the periods from t to u then the total inventory within the system at the end period $t-1$ must be greater than or equal to the total external demand from t to u. Constraints (48) specify that if there is no delivery to node i during the period from t to u then the inventory at node i at the end period $t-1$ must be greater than or equal to the demand of i from periods t to u.

We now explain under what conditions the PRP has Wagner-Whitin costs (WWC). Recall that if the sum of the unit production and holding costs in a period is greater than or equal to the unit production cost in the next period, then the problem has WWC. Since we have assumed that the unit production costs are time-invariant, the WWC assumption holds at the plant level if the holding cost h_0 is non-negative. At the customer level there is no production cost and thus the WWC assumption holds for customer i if $h_i - h_0 \geq 0$ as $h_i - h_0$ is the holding cost for customer i in the echelon stock formulation (see (29)).

3.9 Uncoordinated Approach

In the uncoordinated approach the PRP is solved sequentially in two phases. In the first phase, the uncapacitated single-item lotsizing problem is solved to optimality with the demand at the plant in period t set equal to d_{0t}. The uncapacitated single-item lotsizing problem can be solved optimally in $O(T \log T)$ (see, e.g., Wagelmans et al., 1992). Let $\overline{p}_t$ be the optimal production values in the first phase. In the second phase, the inventory-routing problem, which is modeled using the facility location formulation, is solved to optimality in a similar way as done for the PRP. In this problem the availability of production volume at the plant is fixed by the solution of the first phase, i.e., the following constraint must be satisfied:

$$\sum_{u=1}^{t} \sum_{i \in N_c} r_{iu} \leq \sum_{u=1}^{t} \overline{p}_u \quad \forall t \in T.$$

We call this as the PD approach. For more information about this approach, the reader is referred Chandra and Fisher (1994) and Fumero and Vercellis (1999). However, in these studies the uncoordinated approach is not solved to optimality.

4 Valid Inequalities

We describe two families of valid inequalities for the PRP, 2-matching and generalized comb inequalities, which are adapted from the literature. Adding cuts strengthens a formulation and leads to more efficient solution methods. We will demonstrate the usefulness of these valid inequalities through computational experiments.

A *comb* is a family $\mathcal{C} = (H, T_1, \ldots, T_q)$ of $q+1$ node subsets, where $q \geq 3$ is an odd integer. The node set H is called the *handle* while $T_1, \ldots, T_q$ are called the *teeth*. In addition, these sets must satisfy the following conditions:

- $T_i \cap T_j = \emptyset$ for all $i \neq j$;
- $T_i \backslash H \neq \emptyset$ and $H \cap T_i \neq \emptyset$ for $i = 1, \ldots, q$.

The comb inequality associated with $\mathcal{C}$ is

$$x_t(\sigma(H)) + \sum_{i=1}^{q} x_t(\sigma(T_i)) \leq |H| + \sum_{i=1}^{q} (|T_i| - 1) - (q+1)/2, \tag{49}$$

where $\sigma(S) = \{(i,j) \in E | i \in S, j \in S\}$. These constraints are known to be facets for the traveling salesman problem (see, e.g., Grötschel and Padberg 1979). Fischetti et al. (1997) generalized comb inequalities to the symmetrical generalized traveling salesman problem (GTSP). In GTSP, given a proper partition of $S_1, \ldots, S_u$ of N_c, the objective is to find a minimum cost Hamiltonian cycle such that at least one node in each subset S_i is visited. Node subsets S_i are called clusters. These generalized comb inequalities can be directly used in the PRP by considering the whole node set N as a single cluster. This results in the following proposition.

Proposition 1. *Consider comb $\mathcal{C} = (H, T_1, \ldots, T_q)$. For $i = 1, \ldots, q$ let b_i be any node in $H \cap T_i$, and let o_i be any node in $T_i \backslash H$. Inequalities*

$$x_t(\sigma(H)) + \sum_{i=1}^{q} x_t(\sigma(T_i)) + \sum_{i \in N} \beta_i (1 - z_{it}) \leq |H| + \sum_{i=1}^{q} (|T_i| - 1) - (q+1)/2, \tag{50}$$

where $\beta_j = 0$ for all $j \in N \backslash (H \cup T_1 \cup \ldots \cup T_q)$, $\beta_j = 1$ for all $j \in H \backslash (T_1 \cup \ldots \cup T_q)$, and for $j = 1, \ldots, q$:

- $\beta_j = 2$ *for $j \in T_i \cap H$, $j \neq b_j$;*
- $\beta_{b_j} = 1$;
- $\beta_j = 1$ *for $j \in T_i \backslash H$, $j \neq o_j$;*
- $\beta_{o_j} = 0$;

are valid for the PRP.

If for comb $\mathcal{C}$ the following conditions hold:

- $T_i \cap T_j = \emptyset$ for all $i \neq j$;
- $|T_i \backslash H| = 1$ and $|H \cap T_i| = 1 \neq \emptyset$ for $i = 1, \ldots, q$;

then the inequalities associated with the comb $\mathcal{C}$ are known as *2-matching* inequalities. Therefore, a 2-matching inequality is equivalent to a comb inequality with exactly two nodes in every tooth.

5 Branch-and-Cut Algorithm

In this section, we describe a branch-and-cut algorithm to solve the PRP. We focus on the separation procedures used to identify violated inequalities.

First the LP relaxation of the problem without GSECs is solved. Thus, all the formulations contain the constraints mentioned in their respective sections except the GSECs (8) and the integrality requirements on the variables. If the solution of this relaxation is not an integer feasible solution or if it contains subtours, then the branch-and-cut algorithm is started.

At each node of the branch-and-bound tree, the separation algorithms described in Sections 5.1.1 and 5.1.2 are called to detect the inequalities violated by the current solution. We call the separation algorithm in the following order. First, we call the heuristic separation algorithm for the GSECs (8). Then we call the exact one. Finally, we call the separation algorithm for the generalized comb inequalities (50). If one of these separation algorithms finds the violated inequalities then they are added to the formulation, which is re-optimized, and the separation process is started from the beginning. This procedure is repeated until no violated inequalities are found. For the basic formulation the comb or 2-matching separation algorithms are not called as it serves as a comparison basis for other formulations (i.e., no improvements or strengthenings are done for the basic formulation).

Once no further violated inequalities are detected, branching starts. We use a branching priority rule where we first branch on y variables, next on z variables, and lastly on x variables.

5.1 Separation procedures

We outline the separation algorithms which are applied for each time period to detect the violated valid inequalities. We simplify the notation used in this section by dropping the time period index t and consider the PRP as if it contained only one time period. We denote by $G^* = (N^*, E^*)$ the support graph associated with the given fractional solution (x^*, z^*), where $N^* = \{i \in N | z_i^* > 0\}$ and $E^* = \{(i,j) \in E | x_{ij}^* > 0\}$. We define $x(B : C) = \sum_{i \in B, j \in C} x_{ij}$.

5.1.1 Generalized Subtour Elimination Constraints.

In the heuristic separation algorithm we look for connected components of the support graph G^*. This idea has been used in the context of the TSP (see, e.g., Applegate et al. 2007). If we find more than one connected component, then we detect a set of violated generalized subtour elimination constraints (GSECs) because $x(\delta(S_i)) = 0$ for each connected component S_i. For each connected component we only separate the maximal violated GSEC. If the found connected component S_i consists of the plant node, then we separate the GSEC in the format $x(\delta(S_i)) \geq 2 \max\{z_i^* | i \in N^* \backslash S_i\}$, otherwise we separate the GSEC in the format $x(\delta(S_i)) \geq 2 \max\{z_i^* | i \in S_i\}$. The reason for the former case is that the flow from a connected component that contains the plant does not have to be positive unless there is at least one customer that must be visited and that does not belong to the connected component being considered.

In the exact separation algorithm for each customer node $t \in N^*$ a minimum $s - t$ cut problem is solved where s is a plant node. A minimum $s - t$ cut problem asks for a minimum set of edges whose removal disconnects s from t. Let (S, T) denote the minimum cut where disjoint sets S and T contain nodes s and

t, respectively. If the value of the minimum cut (S,T) is less than $2z_t$, the cut defines a violated GSEC. For each detected GSEC we separate the GSEC in the format $x(\delta(T)) \geq 2\max\{z_i^*|i \in T\}$. We use a minimum $s-t$ cut algorithm of the Concorde callable library (see, e.g., Applegate et al. 2005) that implements the push-relabel flow algorithm described in Goldberg and Tarjan (1988).

5.1.2 Comb and 2-Matching Inequalities.

We develop a heuristic separation algorithm for the comb inequalities. These inequalities were presented in Fischetti et al. (1997) in the context of the symmetric generalized TSP. They proposed a separation algorithm only for the subset of the generalized comb inequalities that are called the generalized 2-matching inequalities. Our separation algorithm is able to detect the violated generalized comb inequalities, not only the violated 2-matching inequalities. Our separation algorithm contains two phases. In the first phase we try to identify the 2-matching inequalities. If we find a 2-matching inequality, then in the second phase, we try to extend the 2-matching inequality to a comb inequality. Next, we describe both phases in detail.

Our separation algorithm for the 2-matching inequalities is very similar to that of Fischetti et al. (1998). Let x_{ij}^* be viewed as a weight associated with each $(i,j) \in E^*$. We apply Kruskal's greedy algorithm (Kruskal 1956) to find a minimum-weight spanning tree on G^*. When this algorithm selects a new edge (i,j) we check in the subgraph of G^* induced by all the edges selected so far, whether this edge forms a cycle or not. If it does, then we consider the connected component that contains both nodes i and j as a handle H. In this way we generate efficiently almost all the connected components H of subgraph $G_\theta^* = (N, E_\theta^*)$ induced by $E_\theta^* = \{(i,j) \in E : 0 < x_{ij}^* < \theta\}$ for every $\theta \in (0,1]$.

For each handle H we first select a subset F of edges that are incident to the set H in the following way:

$$F = \operatorname*{argmax}_{S \subseteq \delta(H)} \left\{ \sum_{(i,j)\in S} x_{ij}^* : \text{ all the edges in } S \text{ are pairwise disjoints} \right\}.$$

Let $F = \{\alpha_1, \ldots, \alpha_u\}$ with $x_{\alpha_1}^* \geq x_{\alpha_2}^* \geq \cdots \geq x_{\alpha_u}^*$. Then we select the first q edges from the set F such that $u \geq q \geq 3$, q is odd, and it maximizes the following expression:

$$x_{\alpha_1}^* + (x_{\alpha_2}^* + x_{\alpha_3}^* - 1) + \cdots + (x_{\alpha_{q-1}}^* + x_{\alpha_q}^* - 1).$$

These q edges are teeth for the handle H. We denote the found 2-matching by $\mathcal{M} = (H, T_1, \ldots, T_q)$.

In the second phase, given a 2-matching $\mathcal{M}$, we use a simple heuristic algorithm to extend the given 2-matching inequality to a comb inequality. For notational reasons we denote here the handle H by T_0. For each node $j \in N^*$ we determine whether it belongs to the subsets $T_0 \setminus \bigcup_{i=1}^q T_i$, $N^* \setminus \bigcup_{i=0}^q T_i$, or $\bigcup_{i=1}^q T_i$. In the first case, we determine an index $\tilde{k}$ for which the following expression is maximized:

$$\max_{k=1,\ldots,q} \{x(T_k, j) - z_j\}.$$

If the expression $x(T_{\tilde{k}}, j) - z_j$ is positive, we extend subset $T_{\tilde{k}}$ as $T_{\tilde{k}} = T_{\tilde{k}} \cup \{j\}$. In the second case, we again determine an index $\tilde{k}$ for which the following expression is maximized:

$$\max_{k=0,\ldots,q} \{x(T_k, j) - z_j\}.$$

If the expression $x(T_{\tilde{k}}, j) - z_j$ is positive, we extend subset $T_{\tilde{k}}$ as $T_{\tilde{k}} = T_{\tilde{k}} \cup \{j\}$. Extensions mean that the value of the left-hand side of the inequality (50) is increased more than the value of the right-hand side. In the last case, we do nothing. This process is repeated until no further extension can be done. After the second phase, we check whether the inequality (50) is violated or not. To separate the 2-matching inequalities only, skip the second phase.

6 A Priori Tour Based Heuristic

We have adapted the effective a priori tour based heuristic of Solyalı and Süral (2008), originally proposed for an inventory-routing problem, to find an integer feasible solution for the PRP. The main idea of the heuristic

is to replace the vehicle routing decisions in the mathematical programming formulation with constraints dictating a global tour (i.e., a priori tour) over all the nodes of the problem. For a given subset of nodes to be visited in a period, the route of the vehicle is automatically found by skipping the non-visited nodes and following the precedence order of the nodes in the a priori tour. To implement the idea, one has to solve a TSP over all the nodes (not necessarily to optimality) to obtain the sets of nodes that can be visited before and after visiting a node i ($i \in N$). These sets are denoted by π_i, and σ_i, respectively. Note that the sets π_i and σ_i involve the plant (i.e., node 0) for all $i \in N_c$, and π_0 and σ_0 involve all $i \in N_c$. This idea was initially introduced by Pinar and Süral (2006) in the context of an inventory-routing problem but within a weak MIP formulation which could not be solved to optimality; thus hiding its effectiveness. Solyalı and Süral (2008) showed the effectiveness of the a priori tour based heuristic by embedding the idea into a strong MIP formulation. We therefore use the FISP formulation to implement the heuristic. We now give the resulting formulation which we refer to as APTF:
Minimize

$$\sum_{t\in T}\left(s \times y_t + \sum_{i\in N} h_i I_{it} + \sum_{i\in N}\sum_{j\in\sigma_i} c_{ij} x'_{ijt}\right) \tag{51}$$

subject to (14), (15), (22) – (28), and

$$\sum_{j\in\sigma_i} x'_{ijt} = z_{it}, \qquad \forall i \in N, \forall t \in T, \tag{52}$$

$$\sum_{j\in\pi_i} x'_{jit} = z_{it}, \qquad \forall i \in N, \forall t \in T, \tag{53}$$

$$x'_{ijt} \in \{0,1\}, \qquad \forall i \in N, \forall j \in \sigma_i, \forall t \in T, \tag{54}$$

where x'_{ijt} is a binary variable taking value 1 iff the vehicle travels to node j immediately after node i. Note that the x'_{ijt} variables above are defined based on the a priori tour. Therefore, the graph associated with this formulation is not the same as in other formulations presented in this paper. It is a directed incomplete graph that consists of an arc (i,j) if for node $i \in N$, it holds that $j \in \sigma_i$. As a consequence, unlike the x_{ijt} variables, the x'_{ijt} variables are now associated with directed arcs instead of edges. Also, note that the degree constraints and the generalized subtour elimination constraints are replaced by constraints (52) and (53) which impose the vehicle to follow the given a priori tour. Although any feasible solution to the above formulation is a valid upper bound for the PRP, we solve it to optimality and improve the objective value by solving a TSP for each $t \in T$ based on the customers visited in t. The computational effectiveness of the heuristic, which we refer to as the APT heuristic, depends on the solvers used for solving the TSPs and the APTF formulation. Thanks to the publicly available efficient TSP solver Concorde, MIP solver CPLEX, and our strong formulation APTF, we are able to obtain the initial upper bound within about one minute even for the largest instances generated (see Section 7).

7 Computational Experiments

In the literature several datasets for the PRP have been published (see Boudia et al. 2007 or Archetti et al. 2009). These sets have mainly been designed for heuristic algorithms, making most instances too large for our exact algorithms. In addition, all of these datasets involve capacity constraints on vehicles. Therefore, we have generated a new set of instances that is more suitable for our purposes. Yet, we use an instance generation scheme very similar to that of Archetti et al. (2009).

We generate instances by varying the following parameters: 1) the interval from which the holding costs h_i, $i \in N_c$, are drawn; 2) the interval from which the coordinates of the nodes are drawn; 3) the number of customers n; 4) the number of time periods m; and 5) the probability that the demand of a customer is equal to zero in each time period. Holding costs are varied to see whether the presence or absence of WWC has an effect. We look at different values for the interval of coordinates to measure the impact of putting more weight on the distribution part (i.e., higher routing costs c_{ij}). The probability that the demand of a customer is equal to zero is varied to see whether the presence of zero demand has an effect. We consider 252 different parameter combinations in total. The first 144 combinations form a basic set and the remaining ones are used only to examine the limit of our formulations and the performance of the APT heuristic. For each combination, which we call a class, we generate 10 random instances, thus yielding a total of 2520 different instances.

For all classes, the setup cost s and the holding cost at the plant h_0 are set to 1000 and 3, respectively, and the demand d_{it} of customer i in period t is randomly selected from the interval $[5, 25]$. The values of parameters for the basic set, classes 1 to 144, are shown in Table 1 while the values for the second set, classes 145 to 252, are shown in Table 2. The first three columns indicate the class identifiers with different upper bounds (CUB) for the intervals from which the coordinates of the nodes are drawn. The lower bound of the interval is equal to 0 for all the classes. For example, for the instances in classes 1, 49, and 97, all the parameters and intervals are the same except the coordinates of the nodes, which are chosen in the square $[0, 500] \times [0, 500]$, $[0, 1000] \times [0, 1000]$, and $[0, 2500] \times [0, 2500]$, respectively. Column HCC indicates the interval from which the holding cost for each customer is randomly selected. Column P gives the probability that the demand of a customer is equal to zero. In other words, when defining the demand of a customer, first a number from $[0, 100]$ is drawn. If it is less than P, then the demand of the customer is set equal to 0, otherwise the demand is drawn from $[5, 25]$. Columns 6 and 7 state the number of customers n and the number of time periods m, respectively. In all cases, random selections are performed according to a uniform distribution using a one decimal accuracy, except demands for which random numbers are selected according to a uniform, integer distribution. The generated instances are available at http://www.sal.tkk.fi/en/personnel/mirko.ruokokoski/.

Table 1: Parameter values used in the generation of classes 1 to 144.

Class Id / CUB			HCC	P	n	m	Class Id / CUB			HCC	P	n	m
500	1000	2500					500	1000	2500				
1	49	97	[1, 3]	0	15	6	25	73	121	[1, 3]	0	30	10
2	50	98	[1, 3]	25	15	6	26	74	122	[1, 3]	25	30	10
3	51	99	[1, 3]	50	15	6	27	75	123	[1, 3]	50	30	10
4	52	100	[1, 3]	75	15	6	28	76	124	[1, 3]	75	30	10
5	53	101	[1, 5]	0	15	6	29	77	125	[1, 5]	0	30	10
6	54	102	[1, 5]	25	15	6	30	78	126	[1, 5]	25	30	10
7	55	103	[1, 5]	50	15	6	31	79	127	[1, 5]	50	30	10
8	56	104	[1, 5]	75	15	6	32	80	128	[1, 5]	75	30	10
9	57	105	[3, 5]	0	15	6	33	81	129	[3, 5]	0	30	10
10	58	106	[3, 5]	25	15	6	34	82	130	[3, 5]	25	30	10
11	59	107	[3, 5]	50	15	6	35	83	131	[3, 5]	50	30	10
12	60	108	[3, 5]	75	15	6	36	84	132	[3, 5]	75	30	10
13	61	109	[1, 3]	0	20	8	37	85	133	[1, 3]	0	40	15
14	62	110	[1, 3]	25	20	8	38	86	134	[1, 3]	25	40	15
15	63	111	[1, 3]	50	20	8	39	87	135	[1, 3]	50	40	15
16	64	112	[1, 3]	75	20	8	40	88	136	[1, 3]	75	40	15
17	65	113	[1, 5]	0	20	8	41	89	137	[1, 5]	0	40	15
18	66	114	[1, 5]	25	20	8	42	90	138	[1, 5]	25	40	15
19	67	115	[1, 5]	50	20	8	43	91	139	[1, 5]	50	40	15
20	68	116	[1, 5]	75	20	8	44	92	140	[1, 5]	75	40	15
21	69	117	[3, 5]	0	20	8	45	93	141	[3, 5]	0	40	15
22	70	118	[3, 5]	25	20	8	46	94	142	[3, 5]	25	40	15
23	71	119	[3, 5]	50	20	8	47	95	143	[3, 5]	50	40	15
24	72	120	[3, 5]	75	20	8	48	96	144	[3, 5]	75	40	15

All formulations and algorithms were implemented in C++ using Concert Technology and were solved by CPLEX 10.11. CPLEX was used with the default setting. We have also used CPLEX callbacks to

Table 2: Parameter values used in the generation of classes 145 to 252.

Class Id / CUB			HCC	P	n	m	Class Id / CUB			HCC	P	n	m
500	1000	2500					500	1000	2500				
145	181	217	[1, 3]	0	40	8	163	199	235	[1, 5]	50	60	8
146	182	218	[1, 3]	25	40	8	164	200	236	[1, 5]	75	60	8
147	183	219	[1, 3]	50	40	8	165	201	237	[3, 5]	0	60	8
148	184	220	[1, 3]	75	40	8	166	202	238	[3, 5]	25	60	8
149	185	221	[1, 5]	0	40	8	167	203	239	[3, 5]	50	60	8
150	186	222	[1, 5]	25	40	8	168	204	240	[3, 5]	75	60	8
151	187	223	[1, 5]	50	40	8	169	205	241	[1, 3]	0	80	8
152	188	224	[1, 5]	75	40	8	170	206	242	[1, 3]	25	80	8
153	189	225	[3, 5]	0	40	8	171	207	243	[1, 3]	50	80	8
154	190	226	[3, 5]	25	40	8	172	208	244	[1, 3]	75	80	8
155	191	227	[3, 5]	50	40	8	173	209	245	[1, 5]	0	80	8
156	192	228	[3, 5]	75	40	8	174	210	246	[1, 5]	25	80	8
157	193	229	[1, 3]	0	60	8	175	211	247	[1, 5]	50	80	8
158	194	230	[1, 3]	25	60	8	176	212	248	[1, 5]	75	80	8
159	195	231	[1, 3]	50	60	8	177	213	249	[3, 5]	0	80	8
160	196	232	[1, 3]	75	60	8	178	214	250	[3, 5]	25	80	8
161	197	233	[1, 5]	0	60	8	179	215	251	[3, 5]	50	80	8
162	198	234	[1, 5]	25	60	8	180	216	252	[3, 5]	75	80	8

implement our separation algorithms. Only the MIP optimality gap was strengthened from 10^{-4} to 10^{-6}. All computational experiments were performed on a workstation with the lx24-amd64 architecture and 2 GB of memory. We report average values for each class, unless otherwise indicated. Computation times are reported in seconds. A CPU time limit of two hours (7200 seconds) was imposed for each instance.

Our computational experiments focus on five aspects. First, we compare the different formulations with each other. Second, we measure the impact of the valid inequalities on the lower bound and the overall solution time. Third, we compare the uncoordinated approach with the coordinated approach. Fourth, we study the limit of our exact algorithms by increasing the number of customers while keeping the number of time periods fixed. Fifth, we examine the performance of the APT heuristic.

7.1 Comparison Between Different Formulations

We begin this section by comparing the different formulations introduced in the paper. These formulations are: basic, four index facility location, four index shortest path, echelon stock facility location, echelon stock shortest path, and echelon stock Wagner-Whitin costs. Results are reported in Tables 3 – 9.

In Table 3 each element $[j,k]$ gives the number of times out of 1440 instances that formulation j is at least 10 percent faster than formulation k, i.e., it holds that $CPU^j < 0.9 \times CPU^k$, where CPU^j is CPU time for formulation j. In each column k the largest number is shown in bold, to point out which formulation j dominates formulation k the most. From this table we see that FIFL leads to the fastest algorithm for most of the instances.

In Table 4 each element $[j,k]$ gives the number of times at which $\underline{Z}^j > \underline{Z}^k + 0.1$ over all the instances, where $\underline{Z}^j$ is the LP relaxation value of formulation j without generalized subtour elimination constraints (8) and without additional cuts (LPGSEC relaxation for short). The purpose of using the constant 0.1 is to eliminate the impact of rounding errors. Similarly, bold elements indicate dominance. From this table we observe that $\underline{Z}^{ESSP} = \underline{Z}^{ESFL} \geq \underline{Z}^{ESWW} \geq \underline{Z}^{WEAK}$, which can be shown as follows. Consider the echelon stock formulations. Each of them contains a set of uncapacitated lotsizing (ULS) subproblems which are reformulated. It is known that for a single item ULS problem the shortest path, the facility location reformulation, and the basic formulation together with (l, S) inequalities provide the same lower bound

(Pochet and Wolsey 2006). Hence, the first equality must hold. As the (l, S, WW) inequalities are only a subset of the general (l, S) inequalities, the second relationship must hold. Note that when the lotsizing problem has Wagner-Whitin costs, the (l, S, WW) inequalities are sufficient to describe the convex hull of the single item ULS problem, and the second relationship becomes an equality. The last inequality comes from the fact that adding (l, S, WW) inequalities to the formulation will lead to a stronger formulation.

Table 4 also suggests that $\underline{Z}^{FISP} \geq \underline{Z}^{FIFL} \geq \underline{Z}^{ESFL}$. Let us consider the one-warehouse multi-retailer (OWMR) problem. This problem is the special case of the PRP in which routing costs are equal to zero, i.e., $c_{ij} = 0$ for each edge (i, j). Solyalı and Süral (2009) prove that the LP relaxation value of the FIFL formulation of the OWMR problem is greater than or equal to the that of ESFL (Theorem 1), and the LP relaxation value of the FISP formulation of the OWMR problem is greater than or equal to that of FIFL (Theorem 2). Since the routing part of our formulations is the same, it follows immediately from Theorems 1 and 2 of Solyalı and Süral (2009) that $\underline{Z}^{FISP} \geq \underline{Z}^{FIFL} \geq \underline{Z}^{ESFL}$.

Table 3: Each element $[j, k]$ in this table gives the number of times formulation j is at least 10 percent faster than formulation k over all instances in classes from 1 to 144.

	WEAK	ESFL	ESSP	ESWW	FIFL	FISP
WEAK	-	447	628	428	328	520
ESFL	850	-	1009	622	**377**	675
ESSP	697	169	-	259	183	353
ESWW	867	326	937	-	349	609
FIFL	**971**	**876**	**1126**	**925**	-	**1206**
FISP	833	571	898	648	77	-

Table 4: Each element $[j, k]$ in this table gives the number of times formulation j provides the better LP relaxation value without generalized subtour elimination constraints and without any valid inequalities compared to formulation k over all instances in classes from 1 to 144.

	WEAK	ESFL	ESSP	ESWW	FIFL	FISP
WEAK	-	0	0	0	0	0
ESFL	**1438**	-	0	123	0	0
ESSP	**1438**	0	-	123	0	0
ESWW	**1438**	0	0	-	0	0
FIFL	**1438**	**1385**	**1385**	**1385**	-	0
FISP	**1438**	**1385**	**1385**	**1385**	**25**	-

Tables 5 – 9 show a different comparison among the formulations over the classes from 1 to 144. Due to limited space, we report aggregate information based on parameter combinations. Combinations are chosen due to our observation that the holding cost interval parameter is the least important while the number of customers and time periods are the most important for CPU time.

In Tables 5 – 9 all the columns are the same except the second one. The first column indicates the number of customers n. In Tables 6 and 7 the second column gives the upper bound (CUB) of the interval from which the coordinates of nodes are drawn whereas in Tables 8 and 9 it gives the probability of zero demand (P) of a customer. Column FORM shows the formulation name. The remaining columns, except the last one, report the average values calculated over the instances with parameters given under n, and CUB or P. These remaining columns indicate the number of nodes (#N), the number of subtour elimination cuts (#SC), the number of 2-matching cuts (#2C), the number of comb cuts (#CC), the lower bound without cuts, i.e., the LP relaxation value calculated at the root node without considering the generalized subtour elimination constraints and additional cuts ($\underline{Z}$), the lower bound value with cuts ($\underline{Z}^C$), and the number of instances solved to optimality (#O). The lower bound with cuts is a lower bound value at the root node after adding cuts generated by our separation algorithms and cuts generated by CPLEX. Note that CPLEX may strengthen the formulation in MIP preprocessing phase before solving the root node. All the lower bound values are normalized with respect to the optimal value (or the best upper bound value if the optimal value is not known). For each parameter combination, the best CPU time and lower bound are shown in bold.

Table 5: Comparison between different formulations over instances with 15, 20, 30, and 40 customers.

n	FORM	CPU	#N	#SC	#2C	#CC	$\underline{Z}$	$\underline{Z}^C$	#O
15	WEAK	0.15	0.3	44.8	0.0	0.0	72.92	99.92	360
	ESFL	0.15	0.6	44.4	0.3	0.0	91.14	99.71	360
	ESSP	0.18	0.7	42.4	0.3	0.0	91.14	99.68	360
	ESWW	0.15	0.5	45.3	0.2	0.0	91.14	99.80	360
	FIFL	**0.09**	0.0	26.8	0.3	0.0	**92.86**	**100.00**	360
	FISP	0.10	0.0	26.8	0.3	0.0	**92.86**	**100.00**	360
20	WEAK	2.03	6.5	223.3	0.0	0.0	61.10	99.39	360
	ESFL	0.97	4.4	189.7	1.9	0.1	87.72	99.39	360
	ESSP	1.36	4.8	188.8	2.0	0.1	87.72	99.35	360
	ESWW	1.03	4.0	186.7	1.8	0.1	87.72	99.46	360
	FIFL	**0.64**	0.4	93.0	1.3	0.0	**90.49**	**99.99**	360
	FISP	0.93	0.5	92.9	1.3	0.0	**90.49**	**99.99**	360
30	WEAK	71.32	755.5	1,095.9	0.0	0.0	51.00	98.77	360
	ESFL	10.74	22.6	837.5	15.4	0.6	84.85	99.10	360
	ESSP	19.16	23.5	832.9	15.1	0.6	84.85	99.09	360
	ESWW	13.41	27.6	822.4	14.2	0.7	84.84	99.14	360
	FIFL	**7.33**	20.7	370.9	12.0	0.2	**88.26**	**99.94**	360
	FISP	9.71	12.9	367.5	11.7	0.3	**88.26**	**99.94**	360
40	WEAK	2,370.03	1,970.9	6,845.2	0.0	0.0	38.73	97.33	312
	ESFL	308.06	348.7	3,649.5	75.1	3.6	82.56	98.85	358
	ESSP	486.76	430.6	3,681.3	75.5	3.9	82.56	98.85	356
	ESWW	397.83	373.4	3,635.7	70.3	3.7	82.56	98.87	357
	FIFL	**296.45**	480.9	1,743.9	57.2	2.2	**86.21**	**99.82**	356
	FISP	364.21	370.2	1,736.8	54.5	2.1	**86.21**	**99.82**	356

In order to obtain an overall view, we report the average results for the classes with 15, 20, 30, and 40 customers and 6, 8, 10, and 15 time periods, respectively. Overall results indicate that FIFL is the fastest. For the more difficult classes (i.e., those having 30 and 40 customers), FIFL provides a speed up of around a factor 10 compared to WEAK. The second best times are run by FISP for the classes with 15, 20, and 30 customers and by ESFL for the most difficult class. Of the five reformulations, ESSP is the slowest on average, but substantially better than WEAK, except for the easiest problem class with 15 customers.

We observe that the two four-index formulations need approximately the same number of GSECs. The three echelon stock reformulations almost double this number, while WEAK needs even more. Also for the generalized 2-matching and comb inequalities, the echelon stock reformulations need more inequalities compared to the four index formulations.

The LP relaxation bound LPGSEC of WEAK is 73% for the easiest problem class, but gradually becomes worse for the more difficult classes. For the most difficult class, with 40 customers and 15 time periods, the average LP relaxation lower bound is only 39%. For the five reformulations, this lower bound becomes worse for the more difficult classes. However, the bound for the reformulations is much better compared to that of WEAK. It varies between 83 and 91% for the echelon stock reformulations and between 86 and 93% for the four index formulations. Even though the LP relaxation lower bound without any cuts is very bad for WEAK, the lower bound $\underline{Z}^C$, i.e., after adding subtour elimination constraints and the cuts generated by CPLEX, is quite close to that of the strong formulations. For the easiest class, it becomes even better than the $\underline{Z}^C$ lower bound of the echelon stock reformulations.

Table 6: Comparison between different formulations over instances with 15 and 20 customers and upper bound of the coordinates interval being equal to 500, 1000, and 2500.

n	CUB	FORM	CPU	#N	#SC	#2C	#CC	$\underline{Z}$	$\underline{Z}^C$	#O
15	500	WEAK	0.26	0.5	68.4	0.0	0.0	64.79	99.72	120
		ESFL	0.22	1.1	64.8	0.6	0.0	91.15	99.43	120
		ESSP	0.26	1.3	59.8	0.5	0.0	91.15	99.41	120
		ESWW	0.21	0.9	65.2	0.5	0.0	91.15	99.59	120
		FIFL	**0.10**	0.0	34.3	0.6	0.0	**93.65**	**100.00**	120
		FISP	0.12	0.0	33.9	0.5	0.0	**93.65**	**100.00**	120
15	1000	WEAK	0.13	0.2	47.1	0.0	0.0	69.69	99.97	120
		ESFL	0.17	0.6	50.4	0.2	0.0	89.98	99.62	120
		ESSP	0.19	0.7	49.2	0.3	0.0	89.98	99.57	120
		ESWW	0.16	0.6	52.0	0.2	0.0	89.98	99.71	120
		FIFL	**0.10**	0.0	31.7	0.2	0.0	**92.21**	**100.00**	120
		FISP	0.12	0.0	32.1	0.2	0.0	**92.21**	**100.00**	120
15	2500	WEAK	**0.06**	0.1	19.0	0.0	0.0	78.90	99.98	120
		ESFL	0.07	0.2	18.1	0.1	0.0	91.89	99.90	120
		ESSP	0.08	0.2	18.3	0.1	0.0	91.89	99.87	120
		ESWW	**0.06**	0.1	18.6	0.1	0.0	91.89	99.96	120
		FIFL	0.07	0.0	14.5	0.1	0.0	**92.90**	**100.00**	120
		FISP	0.07	0.0	14.2	0.1	0.0	**92.90**	**100.00**	120
20	500	WEAK	3.92	13.9	358.8	0.0	0.0	53.06	98.14	120
		ESFL	1.39	7.3	268.2	3.3	0.1	88.52	99.04	120
		ESSP	2.06	7.4	269.1	3.5	0.1	88.52	99.04	120
		ESWW	1.51	6.6	262.5	3.0	0.2	88.52	99.14	120
		FIFL	**0.64**	0.9	107.7	1.9	0.0	**92.12**	**99.97**	120
		FISP	1.02	1.1	108.7	2.1	0.1	**92.12**	**99.97**	120
20	1000	WEAK	1.76	4.6	224.1	0.0	0.0	57.65	99.45	120
		ESFL	1.07	4.7	207.9	1.1	0.0	86.68	99.19	120
		ESSP	1.44	5.4	204.3	1.2	0.0	86.68	99.15	120
		ESWW	1.12	4.2	205.0	1.1	0.0	86.68	99.30	120
		FIFL	**0.69**	0.2	102.4	0.8	0.0	**90.02**	**99.99**	120
		FISP	1.02	0.3	102.9	0.8	0.0	**90.02**	**99.99**	120
20	2500	WEAK	**0.41**	1.0	87.1	0.0	0.0	67.07	99.94	120
		ESFL	0.45	1.2	93.0	1.3	0.0	88.01	99.67	120
		ESSP	0.57	1.5	93.0	1.3	0.0	88.01	99.62	120
		ESWW	0.46	1.2	92.7	1.3	0.0	88.01	99.71	120
		FIFL	0.57	0.1	68.9	1.1	0.0	**90.03**	**100.00**	120
		FISP	0.77	0.1	67.2	1.0	0.0	**90.03**	99.99	120

When considering lower bounds of the strong formulations, we notice that after adding the cuts the gap decreases substantially since the average $\underline{Z}^C$ values are above 98.8%. The four index formulations provide the best lower bound in both cases, with and without additional cuts. For the easiest class, the gap for the four index formulations is nil at the root node and no branching is needed, whereas for the most difficult class the $\underline{Z}^C$ value is still above 99.8%.

Comparing CPU times of the strong formulations with respect to other parameters, we observe that FIFL is the best formulation all the time, except for the instances with higher routing costs c_{ij} as seen in Tables 6 and 7. For the larger instances ESFL is the fastest. It is because when the impact of production decisions increases relative to that of distribution, the reformulations pay off the effort. In the reverse case, the formulations with less number of variables become suitable. WEAK, ESWW or both are the best for the small instances because the problem can often be solved at the root node or by exploring just a few nodes.

Table 7: Comparison between different formulations over instances with 30 and 40 customers and upper bound of the coordinates interval being equal to 500, 1000, and 2500.

n	CUB	FORM	CPU	#N	#SC	#2C	#CC	$\underline{Z}$	$\underline{Z}^C$	#O
30	500	WEAK	174.91	2,159.2	1,722.6	0.0	0.0	43.48	97.03	120
		ESFL	15.47	42.0	1,172.3	23.8	1.1	86.21	98.61	120
		ESSP	28.26	42.9	1,164.7	24.3	0.9	86.21	98.61	120
		ESWW	20.08	56.0	1,138.0	22.2	1.2	86.20	98.65	120
		FIFL	**7.41**	50.0	408.8	18.6	0.5	90.70	**99.93**	120
		FISP	9.71	29.2	398.2	17.8	0.5	**90.71**	99.92	120
30	1000	WEAK	29.97	88.5	1,063.0	0.0	0.0	47.84	98.60	120
		ESFL	11.19	18.1	872.3	15.7	0.5	84.48	98.93	120
		ESSP	20.49	19.3	868.9	14.5	0.6	84.48	98.93	120
		ESWW	14.22	18.5	845.4	13.9	0.6	84.48	98.97	120
		FIFL	**7.49**	10.0	408.1	12.0	0.2	**88.19**	**99.92**	120
		FISP	10.86	7.8	407.0	11.7	0.2	**88.19**	99.91	120
30	2500	WEAK	9.09	18.7	502.0	0.0	0.0	56.36	99.67	120
		ESFL	**5.57**	7.7	468.0	6.5	0.2	84.46	99.43	120
		ESSP	8.72	8.2	465.2	6.5	0.2	84.46	99.41	120
		ESWW	5.93	8.3	483.9	6.4	0.2	84.46	99.46	120
		FIFL	7.10	2.3	295.9	5.6	0.1	**87.21**	**99.96**	120
		FISP	8.55	1.8	297.2	5.4	0.1	**87.21**	**99.96**	120
40	500	WEAK	3,055.84	2,513.9	9,208.5	0.0	0.0	31.72	95.54	101
		ESFL	374.30	561.2	4,593.8	88.0	4.6	84.07	98.33	118
		ESSP	576.58	556.4	4,628.2	85.2	4.6	84.07	98.34	118
		ESWW	442.22	539.2	4,566.0	84.1	4.7	84.07	98.36	118
		FIFL	**252.29**	850.5	1,781.7	62.7	2.5	**88.88**	**99.88**	118
		FISP	317.38	620.1	1,780.9	56.6	2.1	**88.88**	**99.88**	118
40	1000	WEAK	2,406.21	2,404.8	6,452.5	0.0	0.0	36.00	97.14	101
		ESFL	**380.53**	438.0	3,864.1	96.4	4.8	82.48	98.58	120
		ESSP	612.87	689.1	3,882.4	100.9	5.6	82.48	98.58	118
		ESWW	543.31	538.4	3,870.2	89.9	5.2	82.48	98.60	119
		FIFL	413.48	570.6	1,875.3	76.0	3.5	**86.35**	**99.78**	118
		FISP	496.24	470.7	1,880.2	75.9	3.4	**86.35**	**99.78**	118
40	2500	WEAK	1,648.04	994.0	4,874.4	0.0	0.0	43.54	98.23	110
		ESFL	**169.34**	47.0	2,490.5	40.8	1.3	81.94	99.24	120
		ESSP	270.83	46.2	2,533.2	40.4	1.4	81.94	99.25	120
		ESWW	207.96	42.7	2,470.9	36.8	1.1	81.94	99.26	120
		FIFL	223.57	21.5	1,574.8	32.9	0.5	**84.95**	**99.82**	120
		FISP	279.00	19.7	1,549.2	31.0	0.7	**84.95**	**99.82**	120

Results from Tables 8 and 9 show that increasing the probability that a customer demand is equal to zero usually leads to the easier instances, except the instances with 40 customers and the zero demand probability equal to 50%. These are on average slightly harder to solve than the instances with 40 customers and the zero demand probability equal to 0 or 25%.

As it is mentioned at the beginning of this section, it holds that ESWW, ESFL, and ESSP give the same LPGSEC bound for the cases where the WWC assumption holds. This is also observed in our computational experiments. When the WWC assumption does not hold, we observe that, in accordance with the theory, in some cases ESWW gives an inferior LPGSEC bound compared to ESFL and ESSP. However, the differences are very small, and we observe that the average $\underline{Z}$ values are the same up to two decimals for the three formulations, except for a few sets where there is a difference of 0.01. For ESWW we observe that the

Table 8: Comparison between different formulations over instances with 30 customers and the probability of zero demand equal to 0, 25, 50 and 75.

n	P	FORM	CPU	#N	#SC	#2C	#CC	$\underline{Z}$	$\underline{Z}^C$	#O
30	0	WEAK	139.70	2,325.0	1,348.4	0.0	0.0	44.08	98.37	90
		ESFL	13.57	44.0	806.5	26.3	0.9	87.33	99.36	90
		ESSP	21.55	45.5	831.1	26.5	0.9	87.33	99.35	90
		ESWW	18.34	65.3	800.4	25.4	1.3	87.33	99.46	90
		FIFL	**12.50**	65.4	382.1	22.5	0.5	**89.90**	**99.92**	90
		FISP	14.97	39.7	374.1	22.6	0.7	**89.90**	**99.92**	90
30	25	WEAK	104.27	634.3	1,331.0	0.0	0.0	48.79	98.61	90
		ESFL	13.21	21.4	967.5	19.0	0.6	85.41	99.07	90
		ESSP	23.98	21.0	921.7	17.4	0.6	85.41	99.06	90
		ESWW	17.38	20.5	942.0	16.5	0.6	85.41	99.07	90
		FIFL	**8.50**	12.6	390.3	13.8	0.2	**88.98**	**99.93**	90
		FISP	10.52	6.9	384.8	12.7	0.3	**88.98**	**99.93**	90
30	50	WEAK	30.26	51.2	1,023.1	0.0	0.0	53.97	98.99	90
		ESFL	10.72	16.3	898.1	11.6	0.5	82.50	98.82	90
		ESSP	20.32	17.3	905.3	11.9	0.6	82.50	98.81	90
		ESWW	11.83	15.8	883.0	10.4	0.6	82.50	98.82	90
		FIFL	**5.19**	4.0	367.3	8.4	0.2	**86.66**	**99.95**	90
		FISP	8.25	4.1	373.8	8.0	0.1	**86.66**	**99.95**	90
30	75	WEAK	11.05	11.5	681.0	0.0	0.0	63.92	99.52	90
		ESFL	5.47	8.7	678.0	4.5	0.2	82.38	99.06	90
		ESSP	10.77	10.2	673.6	4.5	0.2	82.38	99.04	90
		ESWW	6.11	8.8	664.5	4.4	0.1	82.37	99.08	90
		FIFL	**3.13**	0.9	344.0	3.5	0.1	**86.13**	**99.95**	90
		FISP	5.09	1.0	337.2	3.3	0.1	**86.13**	99.94	90

instances with WWC are easier to solve and take 85.84 seconds while those without WWC take 111.74 seconds on average. For other formulations the difference is smaller.

As a summary we may conclude that FIFL yields in general the best results for our instances. Therefore, we use FIFL as a reference model in the remaining experiments.

7.2 Impact of Valid Inequalities

The purpose of this section is to investigate the impact of the valid inequalities on the lower bound and CPU time. Tables 10 and 11 show the results obtained by running FIFL on the basic set (classes from 1 to 144). The first three columns indicate the number of customers, the zero demand probability, and the upper bound of interval from which the coordinates of the nodes are drawn. The values in the remaining columns are calculated as average values over classes with parameters shown in the first three columns. Column NC indicates the value of LPGSEC relaxation $\underline{Z}$ as a percentage of the upper bound. It is calculated as $100\underline{Z}/\overline{Z}$ where $\overline{Z}$ is the upper bound value (which corresponds to the optimal solution value or the best upper bound value if the optimum is unknown). Column SC shows the amount of gap closed when adding generalized subtour elimination cuts and cuts generated by CPLEX. It is computed as $100(\underline{Z}^{SC} - \underline{Z})/(\overline{Z} - \underline{Z})$ where $\underline{Z}^{SC}$ is the lower bound value found after adding GSECs (in addition to cuts generated by CPLEX), called the subtour lower bound. Note that GSECs are not valid inequalities in the conventional sense since they are necessary for the formulations. Column 2C and CC indicates the amount of gap between the upper bound and the subtour lower bound closed when adding the 2-matching inequalities or adding the 2-matching and comb inequalities, which is computed as $100(\underline{Z}' - \underline{Z}^{SC})/(\overline{Z} - \underline{Z}^{SC})$ where $\underline{Z}'$ is the lower bound value found using the corresponding cuts. The last three columns report the total CPU time to find out the optimal

Table 9: Comparison between different formulations over instances with 40 customers and the probability of zero demand equal to 0, 25, 50, and 75.

n	P	FORM	CPU	#N	#SC	#2C	#CC	$\underline{Z}$	$\underline{Z}^C$	#O
40	0	WEAK	3,756.05	3,423.6	8,073.1	0.0	0.0	32.27	96.61	63
		ESFL	**361.93**	714.9	3,118.1	89.9	3.6	85.44	99.21	89
		ESSP	433.89	724.6	3,150.4	89.7	3.7	85.44	99.21	89
		ESWW	469.91	720.7	3,020.5	81.7	3.4	85.44	99.27	89
		FIFL	396.71	1,110.1	1,411.7	69.0	2.1	**88.16**	**99.85**	89
		FISP	442.34	787.1	1,412.3	65.8	2.6	**88.16**	**99.85**	89
40	25	WEAK	2,514.01	2,705.6	6,810.8	0.0	0.0	36.36	97.29	79
		ESFL	277.00	153.8	3,755.3	78.9	3.2	83.32	98.83	90
		ESSP	467.94	133.9	3,768.2	78.8	3.8	83.32	98.83	90
		ESWW	358.41	135.2	3,790.5	75.2	3.8	83.32	98.83	90
		FIFL	**212.25**	80.8	1,670.9	58.0	1.6	**86.82**	**99.79**	90
		FISP	297.52	90.0	1,666.3	54.7	2.0	**86.82**	**99.79**	90
40	50	WEAK	2,065.52	1,419.7	6,705.8	0.0	0.0	41.96	97.67	82
		ESFL	**422.61**	482.3	4,162.6	92.3	5.4	80.76	98.57	89
		ESSP	740.79	815.3	4,156.4	94.2	6.4	80.76	98.57	87
		ESWW	563.44	588.4	4,159.2	87.1	5.2	80.76	98.56	88
		FIFL	435.79	716.2	2,057.8	75.0	4.4	**85.15**	**99.81**	87
		FISP	515.13	589.8	2,038.2	73.3	3.0	**85.15**	**99.81**	87
40	75	WEAK	1,144.54	334.6	5,790.9	0.0	0.0	50.82	98.30	88
		ESFL	170.68	43.8	3,561.9	39.3	2.0	78.18	98.58	90
		ESSP	304.42	48.5	3,650.1	39.3	1.7	78.18	98.58	90
		ESWW	199.55	49.4	3,572.7	37.0	2.3	78.18	98.58	90
		FIFL	**141.03**	16.3	1,835.3	26.7	0.5	**82.90**	**99.83**	90
		FISP	201.83	13.9	1,830.2	24.2	0.7	**82.90**	99.82	90

solution when using the corresponding cuts. The last row gives the average values over classes presented in the table.

Tables 10 and 11 show that the integrality gap of the LP relaxation at the root node is quite large for many instances. When adding generalized subtour elimination cuts (together with cuts generated by CPLEX) this gap is closed on average by 98 to 99% depending on the size of the instance. The 2-matching inequalities are able close the remaining gap on average by about 40%. The difference between the 2-matching and comb inequalities (the comb cut separation algorithm also finds the 2-matching cuts) is rather small. In Table 10 for small instances with 15 and 20 customers additional cuts, the 2-matching and comb inequalities, have a very small impact on CPU times. However, Table 11 shows that for the larger instances with 30 and 40 customers the use of the 2-matching inequalities decreases the total CPU time by more than 50% compared to the case in which only the generalized subtour elimination cuts and the cuts generated by CPLEX are added. The comb inequalities are able to decrease CPU times by another 12% compared to the 2-matching inequalities, even though they are only marginally better at closing the remaining gap. For some classes the lower bound value after adding the 2-matching inequalities is higher than after adding the comb inequalities. The reason is the cut tolerance (which is the difference between the left and right hand sides of an inequality so that the inequality is considered to be violated) whose values was set to 0.1 to avoid adding too many cuts.

7.3 Uncoordinated Approach

In this section we compare the uncoordinated approach, which is essentially a heuristic algorithm for solving the PRP, with the coordinated approach to measure the benefits of simultaneously optimizing production and distribution decisions. The objective value of the PD approach contains the setup cost from the first

Table 10: Impact of valid inequalities on the lower bound and the CPU time for FIFL.

			Lower Bound				CPU Time		
n	P	CUB	NC	SC	2C	CC	SC	2C	CC
15	0	500	94.82	99.99	33.33	33.33	0.11	0.11	0.11
		1000	93.92	99.86	33.33	33.33	0.11	0.11	0.11
		2500	94.57	100.00	0.00	0.00	0.09	0.10	0.09
	25	500	93.60	99.90	33.33	33.33	0.13	0.13	0.12
		1000	93.18	99.96	16.67	16.67	0.11	0.11	0.11
		2500	94.96	100.00	0.00	0.00	0.06	0.07	0.06
	50	500	93.25	100.00	0.00	0.00	0.10	0.10	0.10
		1000	90.70	100.00	0.00	0.00	0.11	0.11	0.11
		2500	92.60	100.00	0.00	0.00	0.06	0.07	0.06
	75	500	92.40	99.78	33.33	33.33	0.09	0.09	0.09
		1000	90.39	99.95	33.33	33.33	0.07	0.07	0.08
		2500	87.57	100.00	0.00	0.00	0.05	0.06	0.05
20	0	500	93.03	98.80	26.54	25.36	1.09	0.92	0.90
		1000	91.48	99.86	100.00	100.00	0.98	0.96	0.96
		2500	90.79	99.56	95.93	95.93	0.88	0.96	0.87
	25	500	93.48	99.44	91.11	94.86	0.76	0.72	0.69
		1000	90.72	99.65	68.91	68.91	0.81	0.81	0.81
		2500	91.51	99.70	55.70	55.70	0.70	0.78	0.70
	50	500	91.63	99.36	96.58	96.58	0.52	0.53	0.51
		1000	89.23	99.88	82.04	82.04	0.52	0.52	0.52
		2500	89.02	99.48	95.76	95.76	0.48	0.53	0.46
	75	500	89.26	99.37	10.26	10.26	0.46	0.47	0.45
		1000	88.36	99.55	29.77	29.77	0.48	0.49	0.49
		2500	88.63	99.47	90.79	94.18	0.27	0.30	0.27
		Average	91.63	99.73	42.78	43.03	0.38	0.38	0.36

phase plus the routing and holding costs from the second phase. Figure 2 illustrates the percentage gap over the best lower bound value found by the exact algorithms for each class from 1 to 144. The gap curve uses the left y-axis. Percentage values are computed as $100(Z^{PD} - \underline{Z}^{BEST})/\underline{Z}^{BEST}$ where Z^{PD} is the value found by the PD approach and $\underline{Z}^{BEST}$ is the optimal solution value if the instance is solved to optimality, or the best lower bound at the end of the branch-and-cut algorithm otherwise. Figure 2 also shows CPU times, where the CPU time curve uses the right y-axis.

The figure clearly indicates that an uncoordinated approach can increase the cost significantly. On average, the cost increase is 47%, but it can go up to 204.6% (for one instance). The increase is higher for those classes with higher routing costs. To see this, note that the parameters for classes from 1 to 48, from 49 to 96, and from 97 to 144 are the same except the upper bound of the coordinates interval of the nodes, which are 500, 1000, and 2500, respectively. The average cost increase for these three classes are 20.14, 38.14, and 82.72%. In general CPU times for the PD approach are quite short except for some classes. In Table 12 we compare the PD approach with FIFL. From this table we see that the uncoordinated approach is about 33% faster but is not able to find the optimal solutions. Apparently, it finds the optimal solution only 10 times in solving 1440 instances.

Once more we would like to point out that this is the first time that a comparison is made using optimal solutions for both the coordinated and uncoordinated approaches to solve the PRP in the literature. Now we would like to discuss how our results can be compared with previously obtained results in the literature. Chandra and Fisher (1994) and Fumero and Vercellis (1999) computationally show that the cost savings increase if the capacity is not tight. The savings increase if the time horizon is longer, if the number of customers is large, and if there are more items under consideration. For the case with uncapacitated plant

Table 11: Impact of valid inequalities on the lower bound and the CPU time for FIFL.

			Lower Bound				CPU Time		
n	P	CUB	NC	SC	2C	CC	SC	2C	CC
30	0	500	92.41	97.44	51.85	52.95	148.32	15.91	13.68
		1000	89.81	98.15	42.99	42.99	40.61	12.36	11.88
		2500	89.06	98.85	57.09	56.16	11.53	14.84	11.94
	25	500	90.66	97.99	59.93	60.23	32.79	8.34	7.93
		1000	88.18	98.83	43.73	43.73	9.45	9.32	9.28
		2500	88.90	98.60	74.12	74.12	8.77	9.99	8.30
	50	500	89.78	98.78	52.28	53.01	5.52	5.14	4.97
		1000	88.03	98.29	68.97	70.38	7.62	5.10	5.10
		2500	84.67	99.34	66.09	66.09	5.55	6.36	5.51
	75	500	89.01	99.14	57.17	57.17	3.04	3.12	3.04
		1000	85.93	99.26	44.39	44.39	3.74	3.74	3.71
		2500	85.52	98.94	69.78	69.78	2.72	2.93	2.65
40	0	500	90.97	97.85	49.07	49.04	843.50	354.76	345.97
		1000	88.52	97.33	36.46	36.42	1,759.75	856.47	576.56
		2500	86.76	98.00	37.17	37.17	1,000.66	329.93	267.61
	25	500	89.70	98.16	35.62	37.00	668.10	131.74	124.12
		1000	86.71	97.45	29.12	29.52	1,209.44	308.35	228.15
		2500	85.78	97.90	24.83	24.68	681.23	338.46	284.47
	50	500	87.18	98.05	26.08	26.96	696.77	385.00	411.31
		1000	85.14	97.84	22.18	22.20	915.34	693.79	696.81
		2500	84.42	98.48	37.22	37.22	449.14	212.04	199.26
	75	500	85.95	99.16	31.71	32.43	124.79	126.18	127.75
		1000	83.67	98.52	25.96	25.95	155.72	169.25	152.40
		2500	81.51	98.39	31.54	33.72	142.00	147.68	142.93
		Average	87.43	98.36	44.81	45.14	371.92	172.95	151.89

and capacitated vehicle, Chandra and Fisher (1994) obtain cost savings ranging from 3 to 20%, whereas Fumero and Vercellis (1999) report an overall average cost reduction of 10%. Shiguemoto and Armentano (2010) report that the number of customers has the largest influence, whereas the impact of the number of time periods and items is rather small. They report cost reductions ranging from 23% for 30 customers to 59% for 100 customers. In our experiments, average cost reductions for the instances with 15, 20, 30, and 40 customers are 23.56, 24.53, 31.68, and 33.42%, respectively. We report cost reductions that are generally higher than the ones reported in the literature, but these results are in line with observations of Chandra and Fisher (1994) and Fumero and Vercellis (1999) that the benefit of coordination increases if the production and vehicle capacity is increased. When considering the cost reductions with respect to other parameters, we observe that the impact of zero demand probability, ranges from 60.42 to 32.65% for the instances with 0 and 75% zero demand probability, respectively. We also observe that the cost savings decrease if the holding costs at the customers level is higher.

Table 12: Comparison between PD and FIFL. Numbers are calculated over instances in classes from 1 to 144.

Approach	Ave Gap	Max Gap	Tot CPU time	#O
FIFL	0.000	0.147	30.45 h	1436
PD	47.00	204.6	20.34 h	10

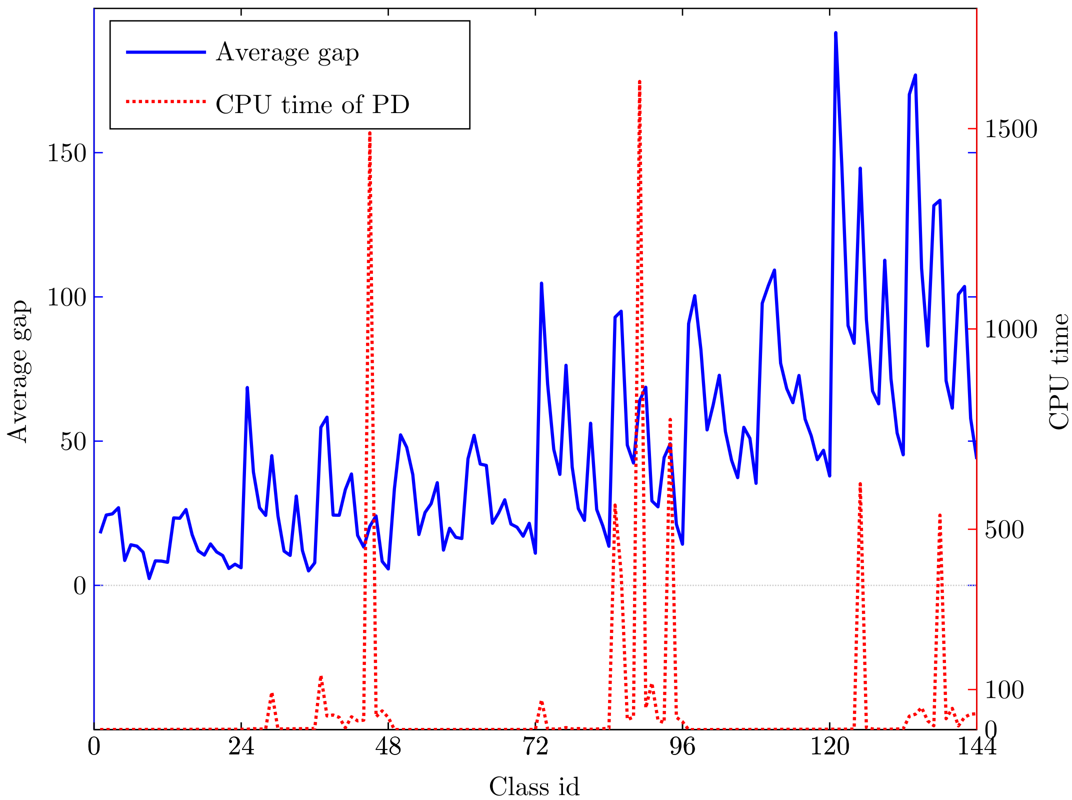

Figure 2: Performance of the PD approach. CPU times are given in seconds and average gaps in percentages over the best lower bound for each class from 1 to 144.

7.4 Limit of the Exact Algorithm

In order to explore the limits of our exact algorithm in terms of capability in solving larger instances, we solved all the classes from 145 to 252 using FIFL, which turned out to be the best formulation in our experiments. In these classes the number of time periods is 8 and the number of customers is either 40, 60, or 80 (see Table 2). The results are reported in Table 13 whose columns are the same as those in Table 5. The results indicate that the algorithm is able to solve the problems with up to 80 customers when there are 8 time periods. We also observe that putting more weight on the distribution decisions makes the problem easier to solve.

Table 13: Results for classes from 145 to 252 for FIFL.

n	CUB	CPU	#N	#SC	#2C	#CC	$\underline{Z}$	$\underline{Z}^C$	#O
40	500	9.01	45.1	665.1	34.7	1.1	90.00	99.89	120
	1000	9.05	12.0	567.0	16.6	0.4	87.46	99.91	120
	2500	6.54	3.3	360.4	9.2	0.3	87.48	99.94	120
60	500	275.94	1,243.0	2,731.7	150.6	6.9	89.01	99.83	117
	1000	289.69	857.9	2,458.0	137.7	5.2	86.68	99.76	117
	2500	53.98	33.3	1,289.9	45.3	0.9	86.53	99.85	120
80	500	2,546.53	2,233.2	9,169.4	408.7	16.6	88.11	99.72	107
	1000	1,464.28	483.2	6,442.8	215.9	6.1	85.84	99.73	119
	2500	705.62	114.0	3,331.6	93.4	2.4	85.09	99.80	120

Table 14: Comparison between APT and FIFL. Numbers are calculated over instances in classes from 1 to 252.

Approach	Ave Gap	Max Gap	Tot CPU time	#O
FIFL	0.000	0.147	209.14 h	2496
APT	0.334	4.954	1.490 h	1058
APT2	0.774	7.397	1.483 h	725

7.5 Performance of the APT Heuristic

Since the APT heuristic turned out to be very fast, we solved each instance in classes from 1 to 252. The gap is calculated as $100(Z^{APT} - \underline{Z}^{BEST})/\underline{Z}^{BEST}$, where Z^{APT} is the solution value found by APT and $\underline{Z}^{BEST}$ is the optimal solution value if the instance is solved to optimality, or the best lower bound at the end of the branch-and-cut algorithm otherwise. Figure 3 depicts the percentage gap. The gap curve uses the left y-axis. Figure 3 also depicts CPU times in seconds of the APT heuristic for each class. The CPU time curve uses the right y-axis. The instances with the largest number of time periods have the largest CPU times while instances with the large number of customers have the largest cost increase. When the zero demand probability is increased, the average CPU time is decreased but the average gap is increased.

In Table 14 we compare the APT heuristic with FIFL. We report values for the APT heuristic without the improvement step (APT2) as well. In this improvement step, we optimally solve a TSP over the customers visited in each time period. The first column indicates the approach used. Columns 2 and 3 give the average and maximum percentage gaps between the solutions found by the approach and $\underline{Z}^{BEST}$, i.e., the best lower bound. Column 4 indicates the total CPU time in hours needed to solve all the instances. Column 5 gives the number of instances that were solved to optimality over 2520 test instances in total. These results show that the APT heuristic performs very well compared to our exact algorithms. The average CPU time is less than 0.8% of the CPU time of FIFL while the average cost increase is only 0.33%. In addition, 42.4% of the solutions found by the APT heuristic are the optimal solutions. When comparing the APT and APT2 heuristics we see that the improvement step takes a very short amount of time, on average 0.01s for each instance, and decreases the remaining gap by 56.8%. We also notice that the APT finds 45.9% more optimal solutions than the APT2.

8 Conclusion

In this paper we study the production-routing problem, consisting of two well-known subproblems in the literature, the lotsizing problem and the inventory-routing problem. We introduce several strong formulations and a branch-and-cut algorithm for the problem.

Our computational experiments show that the four index formulations yield the best lower bounds both with and without cuts (i.e., the LP relaxation without generalized subtour elimination constraints). Experiments on larger instances indicate that the ESFL formulation provides the shortest CPU times for instances in which the distribution subproblem has more impact on the solution relative to production subproblem, whereas FIFL runs the fastest for other instances. The weak formulation is clearly outperformed by the strong formulations.

The experiments concerning valid inequalities show that the addition of the generalized 2-matching and generalized comb inequalities improves the lower bound and decreases CPU times significantly. The APT heuristic finds excellent solutions within a very short amount of time.

In the PRP literature many studies measure the cost savings attainable by a coordinated approach over an uncoordinated one. Nevertheless, none of them uses an exact algorithm. In this paper we measure the impact of coordination in the PRP. Our computational experiments reveal that the coordination of the production and distribution subproblems can lead to very significant savings when compared to the uncoordinated approach.

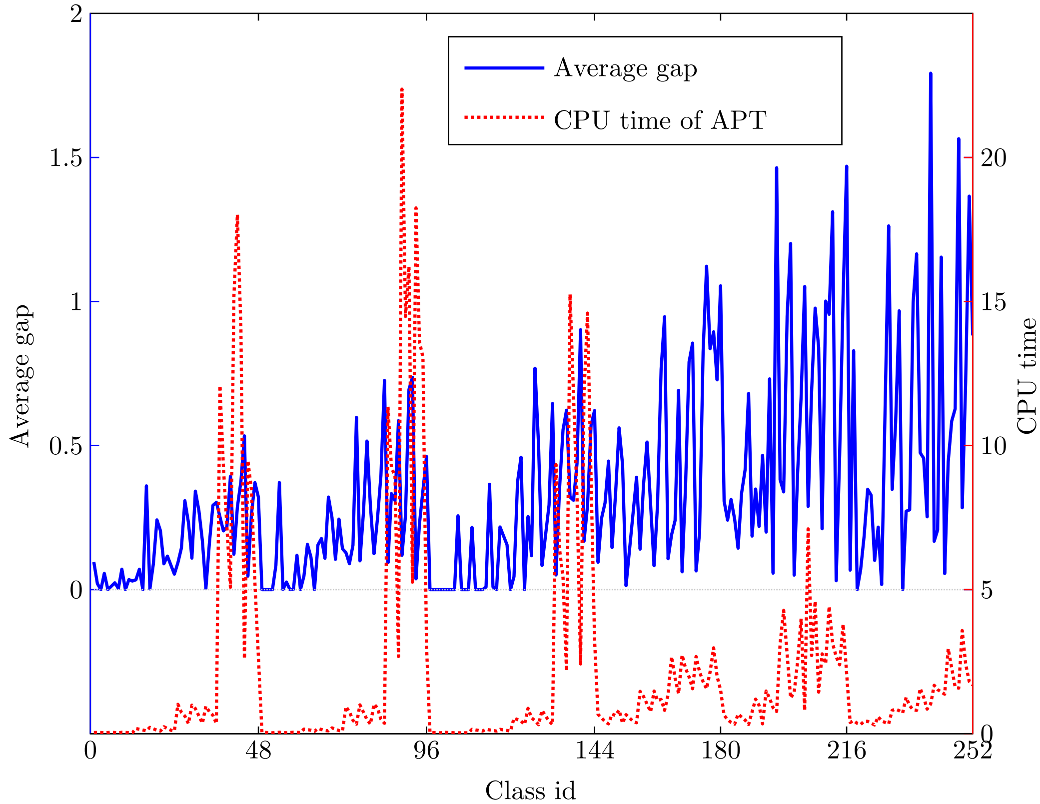

Figure 3: Performance of the APT heuristic. CPU times are given in seconds and average gaps in percentage over the best known lower bound for each class from 1 to 252.

We may conclude that our strong formulations together with a branch-and-cut algorithm using valid inequalities make it possible to solve rather large instances (instances with 8 (respectively 15) time periods and up to 80 (respectively 40) customers) to optimality but larger instances remain a tough challenge.

An interesting topic for future research would be to apply the improved formulations and the APT heuristic to extensions of the PRP problem such as the multi-vehicle case or the problem with capacity constraints for the vehicles and for the plant. Another topic of future work would be to generalize other valid inequalities from the traveling salesman problem such as clique tree and path inequalities.

References

Andersson, H., A. Hoff, M. Christiansen, G. Hasle, A. Løkketangen. 2010. Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research* **37** 1515–1536.

Applegate, D., R. Bixby, V. Chvátal, W. Cook. 2005. Concorde: A code for solving traveling salesman problems. http://www.tsp.gatech.edu/concorde.html.

Applegate, D., R. Bixby, V. Chvátal, W. Cook. 2007. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, Princeton, NJ.

Archetti, C., L. Bertazzi, G. Paletta, M. G. Speranza. 2009. Analysis of the maximum level policy in a production-distribution system. Tech. Rep. 325, University of Brescia.

Barany, I., T.J. Van Roy, L.A. Wolsey. 1984. Uncapacitated lot sizing: the convex hull of solutions. *Math. Programming Stud.* **22** 32–43.

Bard, J. F., N. Nananukul. 2009a. Heuristics for a multiperiod inventory routing problem with production decisions. *Computers & Industrial Engineering* **57** 713–723.

Bard, J. F., N. Nananukul. 2009b. The integrated production-inventory-distribution-routing problem. *Journal of Scheduling* **12** 257ï¿½–280.

Bard, J. F., N. Nananukul. 2010. A branch-and-price algorithm for an integrated production and inventory routing problem. *Computers & Operations Research* **37** 2202–2217.

Bertazzi, L., G. Paletta, M. G. Speranza. 2005. Minimizing the total cost in an integrated vendor-managed inventory system. *Journal of Heuristics* **11** 393–419.

Blumenfeld, D. E., L. D. Burns, C. F. Daganzo, M. C. Frick, R. W. Hall. 1987. Reducing logistics costs at General Motors. *Interfaces* **17**(1) 26–47.

Boudia, M., M. A. O. Louly, C. Prins. 2007. A reactive GRASP and path relinking for a combined production-distribution problem. *Computers & Operations Research* **34** 3402–3419.

Boudia, M., M. A. O. Louly, C. Prins. 2008. Fast heuristics for a combined production planning and vehicle routing problem. *Production Planning & Control* **19** 85–96.

Boudia, M., C. Prins. 2009. A memetic algorithm with dynamic population management for an integrated production-distribution problem. *Eur. J. Oper. Res.* **195** 703–715.

Brown, G., J. Keegan, B. Vigus, K. Wood. 2001. The Kellogg company optimizes production, inventory, and distribution. *Interfaces* **31**(6) 1–15.

Çetinkaya, S., H. Üster, G. Easwaran, B. B. Keskin. 2009. An integrated outbound logistics model for Frito-Lay: Coordinating aggregate-level production and distribution decisions. *Interfaces* **39**(5) 460–475.

Chandra, P. 1993. A dynamic distribution model with warehouse and customer replenishment requirements. *Journal of the Operational Research Society* **44** 681–692.

Chandra, P., M. L. Fisher. 1994. Coordination of production and distribution planning. *Eur. J. Oper. Res.* **72** 503–517.

Clark, A. J., H. Scarf. 1960. Optimal policies for multi-echelon inventory problems. *Management Sci.* **6** 475–490.

Eppen, G. D., R. K. Martin. 1987. Solving multi-item capacitated lot-sizing problems using variable redefinition. *Oper. Res.* **35** 832–848.

Federgruen, A., M. Tzur. 1999. Time-partitioning heuristics: Application to one warehouse, multi-item, multi-retailer lot-sizing problems. *Naval Research Logistics* **46** 463–486.

Fischetti, M., J. J. Salazar González, P. Toth. 1997. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. *Oper. Res.* **45** 378–394.

Fischetti, M., J. J. Salazar González, P. Toth. 1998. Solving the orienteering problem through branch-and-cut. *INFORMS J. Comput.* **10** 133–148.

Fumero, F., C. Vercellis. 1999. Synchronized development of production, inventory, and distribution schedules. *Transportation Sci.* **33** 330–340.

Goldberg, A., R. Tarjan. 1988. A new approach to the maximum-flow problem. *Journal of the ACM* **35** 921–940.

Grötschel, M., M. W. Padberg. 1979. On the symmetric travelling salesman problem I: Inequalities. *Math. Programming* **16** 265–280.

Jans, R., Z. Degraeve. 2008. Modeling industrial lot sizing problem: a review. *International Journal of Production Research* **46** 1619–1643.

Krarup, J., O. Bilde. 1977. Plant location, set covering and economic lot sizes: An $O(mn)$-algorithm for structured problems. L. Collatz et al., ed., *Optimierung bei graphentheoretischen and ganzzahligen probleme*. Birkhauser Verlag, 155–180.

Kruskal, J. B. 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society* **7** 48–50.

Lee, H. L., C. Billington. 1995. The evolution of supply-chain-management models and practice at Hewlett-Packard. *Interfaces* **25**(5) 42–63.

Lei, L., S. Liu, A. Ruszczynski, S. Park. 2006. On the integrated production, inventory, and distribution routing problem. *IIE Transactions* **38** 955–970.

Levi, R., R. Roundy, D. Shmoys, M. Sviridenko. 2008. A constant approximation algorithm for the one-warehouse multiretailer problem. *Management Sci.* **54** 763–776.

Pinar, O., H. Süral. 2006. Coordinating inventory and transportation in vendor managed systems. R. Meller et al., ed., *Proceedings of Material Handling Research Colloquium*. 459–474.

Pochet, Y., L. A. Wolsey. 2006. *Production Planning by Mixed Integer Programming (Springer Series in Operations Research and Financial Engineering)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Shiguemoto, A. L., V. A. Armentano. 2010. A tabu search procedure for coordinating production, inventory and distribution routing problems. *International Transactions in Operational Research* **17** 179–195.

Solyalı, O., H. Süral. 2008. A branch-and-cut algorithm using a strong formulation and an a priori tour based heuristic for an inventory-routing problem. Tech. Rep. 08-07, Middle East Technical University.

Solyalı, O., H. Süral. 2009. The one-warehouse multi-retailer problem: Reformulation, classification, and computational results. Tech. Rep. 09-03, Middle East Technical University.

Stenross, F. M., G. J. Sweet. 1991. Implementing an integrated supply chain. *Annual Conference Proceedings, Oak Brook, Ill: Council of Logistics Management, vol. 2*. 341–351.

Wagelmans, A., S. Van Hoesel, A. Kolen. 1992. Economic lot sizing: An $O(n \log n)$ algorithm that runs in linear time in the Wagner-Whitin case. *Oper. Res.* **40** 145–156.