RAIRO-Oper. Res. 47 (2013) 429–443
DOI: 10.1051/ro/2013050

RAIRO Operations Research
www.rairo-ro.org

LAGRANGEAN HEURISTIC FOR A MULTI-PLANT LOT-SIZING PROBLEM WITH TRANSFER AND STORAGE CAPACITIES

Samuel Deleplanque[1], Safia Kedad-Sidhoum[2] and Alain Quilliot[1]

Abstract. The paper addresses a multi-item, multi-plant lot-sizing problem with transfer costs and capacity constraints. The problem is reformulated according to a multi-commodity flow formalism, and decomposed, through Lagrangean relaxation, into a master facility location problem and a slave minimal cost multi-commodity flow problem. The decomposition framework gives rise in a natural way to designing a Lagrangean based heuristic. Numerical experiments showing the efficiency of the proposed approach are reported.

Keywords. Lot-sizing, multi-plant, storage and transfer capacity, facility location, multi-commodity flow, Lagrangean relaxation.

Mathematics Subject Classification. 90B05.

1. Introduction

Lot-sizing models are production planning models which involve clustering effects: demands may be produced together as part of lots or batches while sharing some kinds of fixed cost resources. This class of problems may be set at both operational and tactical levels. Models may be dynamic or static, and involve assembly processes (multi-level), one or several plants (multi-plant), one or several kinds

Received October 7, 2013. Accepted October 15, 2013.

[1] LIMOS, Bat. ISIMA, Université Blaise Pascal, Campus des Cézeaux, BP 125, 63173 Aubiere. deleplanque.samuel@gmail.com

[2] LIP6, Université Paris 6, 4 place Jussieu 75252 Paris Cedex 05.

Article published by EDP Sciences © EDP Sciences, ROADEF, SMAI 2013

https://doi.org/10.1051/ro/2013050 Published online by Cambridge University Press

of products (multi-item). Performance criteria may be related to production cost minimization, delay or shortage minimization.

One of the major difficulties in solving lot-sizing problems arises from capacity restrictions in each time period. Problems are polynomially solvable when capacities are not considered, as well as when production capacities are constant over the planning horizon [4]. When capacities are allowed to vary over the planning horizon, the problem becomes NP-Hard [3].

Real lot-sizing problems are usually constrained by tight capacity restrictions, high setup times and specific industrial constraints. The classical Multi-item Capacitated Lot-Sizing problem (MCLS) is widely the most basic big-bucket model studied in the context of multi-item capacitated lot-sizing [7]. Nevertheless, obtaining optimal and sometimes even feasible solutions remains challenging. Chen and Thizy [3] prove that the MCLS problem is strongly NP-Hard. Florian et al. [5] and Bitran and Yanasse [2] show that the single-item capacitated lot-sizing problem is NP-hard in the ordinary sense. There are some extensions of the basic lot-sizing problem that can be used to model a variety of industrial problems. We refer to the extensive literature review of Jans and Degraeve [6] for an overview of the latest modeling developments in lot-sizing field.

The problem addressed in this paper is a multi-plant multi-item capacitated lot-sizing problem where transferring some productions between plants is allowed. The items can be directly sent to the customers from the plant where they are produced. A plant can get the items from other plants and keep them in a storage area. The problem is clearly an NP-hard problem. We can refer to the work of Sambivasan and Yahya [9] that describes some Lagrangean based heuristics to solve a relaxed version of the addressed problem and to the work of Nascimento *et al.* [8] that develop a greedy randomized adaptive search heuristic as well as a path-relinking intensification procedure to find cost-effective solutions for the same relaxed version of the problem.

The contribution of the present work is to present a new decomposition scheme of the problem into a master facility location problem and a slave minimum cost multi-commodity flow problem. Since both problems have been widely studied, with many applications to the design of telecommunication networks or to VLSI conception, this kind of decomposition framework open the way to faster generic software development. Part of the difficulty consists in designing the projection part of the decomposition scheme, in such a way that it ensures the feasibility of both capacity and demand constraints. The rest of the paper describes the algorithms which implement the decomposition scheme and tackle the feasibility problem in a Lagrangean relaxation framework. Numerical experiments are reported.

The structure of the paper is the following. Section 2 describes an aggregated formulation of the problem. A reformulation follows in Section 3 based on multi-commodity flows. Section 4 provides the description of the Lagrangean based heuristic to solve the problem that requires solving a facility location problem.

https://doi.org/10.1051/ro/2013050 Published online by Cambridge University Press

Computational experiments are reported in Section 5 before a concluding part in Section 6.

2. Problem definition and mathematical formulation

We consider a set $\mathcal{K}$ of K items (products) that must be produced at I plants over a discrete planning horizon of T periods. We denote by $\mathcal{I}$, the set of plants and $\mathcal{T}$ the set of time periods. A plant i in $\mathcal{I}$ may be at the same time a producer plant when some items are produced at i and a customer plant if i requires some items to meet a given demand. A plant i may also be used as an inventory place for the transfer of items from some producer sites to some customer sites. Moreover, we have to satisfy a demand d_{it}^k for each period t in $\mathcal{T}$, item k in $\mathcal{K}$, and plant i in $\mathcal{I}$. We assume that the productions and transfers are performed within a time period. Indeed, if an item is produced at plant i at period t and transferred to a plant j, $j \neq i$, the transfer occurs at period t. However, if the item has to be stored in plant j, we consider that this storage transition takes place between period t and period $t+1$.

The production of an item k at plant i at period t induces a unit production cost p_{it}^k as well as a fixed setup cost q_{it}^k. A unit holding (storage) cost h_{it}^k is induced when one unit of item k is stored at plant i at period t. A cost c_{ijt}^k is induced for each unit of item k transferred from plant i to plant j at period t. The objective is to minimize the total cost, *i.e.* production, setup, transfer and storage costs to satisfy the demands over the planning horizon.

Capacity constraints are considered on both production, transfer and storage resources. We denote by α_{it}^k the amount of production capacity consumed per unit of item k at plant i at period t. Similarly, β_{it}^k is the fixed (setup) production capacity consumed for producing item k at plant i at period t. For each unit of item k, γ_{it}^k denotes the unit storage consumption at plant i for period t. We assume that the available production (resp. storage) capacity at plant i at period t is A_{it} (resp. B_{it}). The transfer operations must respect capacity restrictions. Each unit of item k transferred from plant i to plant j at period t consumes τ_{ijt}^k. The total consumption induced by the transfer of items from plant i to plant j at period t must not exceed BT_{ijt}.

The problem consists in integrating both production planning decisions and transfer/storage operations. To this end, we propose a Mixed Integer Programming (MIP) model. The decision variables are defined for every i, j in $\mathcal{I}$, k in $\mathcal{K}$, and t in $\mathcal{T}$ as follows:

- variable x_{it}^k represents the (positive) quantity of item k produced at plant i at period t;
- variable z_{it}^k is a binary variable equal to 1 if item k is produced at plant i at period t or 0 otherwise;
- s_{it}^k represents the (positive) quantity of item k stored in plant i at the end of period t (we assume that $s_{i0}^k = 0$);

https://doi.org/10.1051/ro/2013050 Published online by Cambridge University Press

- y^k_{ijt} gives the (positive) quantity of item k transferred from plant i to plant j at period t.

The multi-item Multi-plant Lot-sizing problem under Storage, Transfer and Production Capacity constraints (MLS-STPC problem) can be formulated as follows:

$$\begin{aligned}
\min \quad & \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}(p^k_{\mathrm{it}}x^k_{\mathrm{it}}q^k_{\mathrm{it}}z^k_{\mathrm{it}} + h^k_{\mathrm{it}}s^k_{\mathrm{it}}) \\
& + \sum_{i,j\in\mathcal{I}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} c^k_{\mathrm{ijt}}y^k_{\mathrm{ijt}} && (2.1)
\end{aligned}$$

s.t.

$$x^k_{\mathrm{it}} + s^k_{it-1} + \sum_{j\in\mathcal{I}} y^k_{jit} = d^k_{\mathrm{it}} + s^k_{\mathrm{it}} + \sum_{j\in\mathcal{I}} y^k_{\mathrm{ijt}}, \quad \forall i,k,t \qquad (2.2)$$

$$\sum_{k\in\mathcal{K}} \alpha^k_{\mathrm{it}}x^k_{\mathrm{it}} + \beta^k_{\mathrm{it}}z^k_{\mathrm{it}} \leq A_{\mathrm{it}}, \quad \forall i,t \qquad (2.3)$$

$$\sum_{k\in\mathcal{K}} \gamma^k_{\mathrm{it}}s^k_{\mathrm{it}} \leq B_{\mathrm{it}}, \quad \forall i,t \qquad (2.4)$$

$$\sum_{k\in\mathcal{K},i\in\mathcal{I}} \tau^k_{\mathrm{ijt}}y^k_{\mathrm{ijt}} \leq BT_{\mathrm{ijt}}, \quad \forall j,t \qquad (2.5)$$

$$x^k_{\mathrm{it}} \leq \sum_{l=1}^{N}\sum_{r=t}^{T} d^k_{lr}z^k_{\mathrm{it}}, \quad \forall i,k,t \qquad (2.6)$$

$$x^k_{\mathrm{it}}, s^k_{\mathrm{it}}, y^k_{\mathrm{ijt}} \geq 0, \quad \forall i,j,k,t \qquad (2.7)$$

$$z^k_{\mathrm{it}} \in \{0,1\}, \quad \forall i,k,t. \qquad (2.8)$$

The objective function (2.1) minimizes the total production (fixed and variable), transfer and storage costs. The inventory balance of the quantity of item k during period t at plant j is given by Constraints (2.2). The production (resp. storage) capacity constraint at plant i for period t is given by (2.3) (resp. (2.4)). Similarly, the transfer capacity constraint when transferring items from plant i to plant j at period t is given by (2.5). Finally, Constraints (2.6) are the linking constraints.

In the following section, we propose a different formulation of the MLS-STPC problem based on a multi-commodity reformulation.

3. Multi-commodity flow reformulation of the MLS-STPC problem

The MLS-STPC problem can be formulated as a fixed charge network flow problem. A classical way to tighten this kind of formulation for standard lot-sizing problems is to decompose the flow along each arc of the network as a function of its destination. This defined a so-called multi-commodity formulation assigning a different commodity to each destination node. The decomposition by commodity allows to tighten the formulation by decreasing the upper bounds in the variable upper bound constraints.

Several works in the literature deal with the multi-commodity version of the classical multi-item capacitated lot-sizing problem, we quote the work of Barany *et al.* [1] that proposes exact methods based on the multi-commodity formulation for the problem without setup times. Wu and Golbasi [15] propose a survey

https://doi.org/10.1051/ro/2013050 Published online by Cambridge University Press

paper for multi-item, multi-facility supply chain planning. They propose multi-commodity flow formulations where each commodity is related to the product.

We define a reformulation of the MLS-STPC problem based on network and multi-commodity flows, which will be useful to propose a decomposition of the problem. A commodity will be related to an item in our formulation. We first describe the overall network structure.

We consider a network $G = (V, A)$ associated to the MLS-STPC problem such that the set of nodes is defined by $V = \{(i,t), i \in \mathcal{I}, t \in \mathcal{T}\} \cup \{T+1, 0\}$ where 0 and $T+1$ are two dummy nodes that represent respectively a sink and a source node. The set of arcs will consist in what will be defined as the storage, production, transfer and customer arcs. We define the following set of arcs:

- the set A_s of storage arcs $((i,t),(i,t+1))$ that define the total inventory at plant i at the end of period t, for $i = 1, \ldots, N,\ t = 1, \ldots, T-1$;
- the set A_t of transfer arcs defined by $((i,t),(j,t)), t \in \mathcal{T}, i, j \in \mathcal{I}, i \neq j$, that represent the transferred quantities between plants i and j;
- the set A_p of production arcs defined by $(0,(i,t)), i \in \mathcal{I}, t \in \mathcal{T}$, that define the produced quantities at plant i at period t;
- the set A_c of customer arcs defined by $((i,t), T+1), i \in \mathcal{I}, t \in \mathcal{T}$, that define the quantities consumed at plant i at period t;
- an additional arc $(T+1, 0)$ called the equilibrium arc will be also defined.

The set of arcs A of G is thus defined as follows:

$$A = A_s \cup A_t \cup A_p \cup A_c \cup \{(T+1,0)\}.$$

We now characterize arc labels in the multi-commodity network. Each arc a of A is characterized by a flow, a cost structure and a capacity. Two cost components will be defined for each arc a: a fixed cost $FC^k(a)$ and a variable cost $VC^k(a)$ for each item $k \in \mathcal{K}$ as follows:

- if $a = ((i,t),(i,t+1))$ is a storage arc, then $FC^k(a) = 0$ and $VC^k(a) = h^k_{it}$;
- if $a = ((i,t),(j,t))$ is a transfer arc, then $FC^k(a) = 0$ and $VC^k(a) = c^k_{ijt}$;
- if $a = (0,(i,t))$ is a production arc, then $FC^k(a) = q^k_{it}$ and $VC^k(a) = p^k_{it}$;
- if $a = ((i,t), T+1))$ is a customer arc, then $FC^k(a) = 0$ and $VC^k(a) = 0$;
- if $a = (T+1, 0)$, then $FC^k(a) = 0$ and $VC^k(a) = 0$.

Similarly, we define a fixed $FCP^k(a)$ and variable $VCP^k(a)$ capacity consumption for each arc a of A regarding each item k as follows:

- if $a = ((i,t),(i,t+1))$ is a storage arc, then $FCP^k(a) = 0$ and $VCP^k(a) = \gamma^k_{it}$;
- if $a = ((i,t),(j,t))$ is a transfer arc, then $FCP^k(a) = 0$ and $VCP^k(a) = \tau^k_{ijt}$;
- if $a = (0,(i,t))$ is a production arc, then $FCP^k(a) = \beta^k_{it}$ and $VCP^k(a) = \alpha^k_{it}$;
- if $a = ((i,t), T+1)$ is a customer arc, then $FCP^k(a) = 0$ and $VCP^k(a) = 0$;
- if $a = (T+1, 0)$, then $FCP^k(a) = 0$ and $VCP^k(a) = 0$.

https://doi.org/10.1051/ro/2013050 Published online by Cambridge University Press

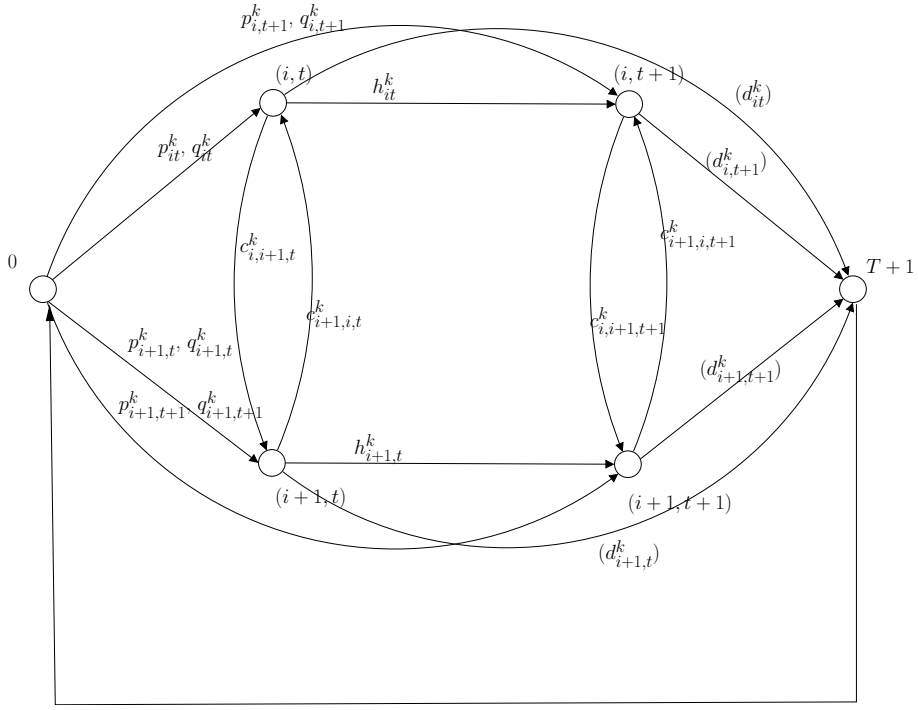


FIGURE 1. An example of the MLS-STPC problem with two plants and two time periods.

The capacity $CA(a)$ of an arc a can be defined as follows:

- if $a = ((i,t),(i,t+1))$ is a storage arc, then $CA(a) = B_{\mathrm{it}}$;
- if $a = ((i,t),(j,t))$ is a transfer arc, then $CA(a) = BT_{\mathrm{ijt}}$;
- if $a = (0,(i,t))$ is a production arc, then $CA(a) = A_{\mathrm{it}}$;
- if $a = ((i,t),T+1)$ is a customer arc, then $CA(a) = +\infty$;
- if $a = (T+1,0)$, then $CA(a) = +\infty$.

In the sequel, we will denote d^k_{it} by $D^k(a)$ if $a = ((i,t),T+1)$, *i.e.* a is a customer arc.

Figure 1 illustrates an instance of the problem for a single commodity k with two plants i and $i+1$ and a planning horizon consisting in periods t and $t+1$. The values between parenthesis represent the demands all the other values are the cost parameters. The zero values are omitted.

In the sequel, we will denote by f a multi-commodity flow and $\mathrm{Ind}(f)$ the boolean vector indexed on the arcs where for $a \in A$, $\mathrm{Ind}(f(a)) = 1$ if $f(a) > 0$ and 0 otherwise. Moreover, we will denote by (a^-, a^+) each arc a, where a^- and a^+

https://doi.org/10.1051/ro/2013050 Published online by Cambridge University Press

represent the endvertices of a, and by A_{pst} (resp. A_{st}) the set $A_p \cup A_s \cup A_t$ (resp $A_s \cup A_t$).

The reformulation of the MLS-STPC problem is:

$$\min \quad \sum_{k\in\mathcal{K}, a\in A_p} FC^k(a)\text{Ind}(f^k(a)) + \sum_{k\in\mathcal{K}, a\in A_{\text{pst}}} VC^k(a)f^k(a) \tag{3.1}$$

$$\sum_{k\in\mathcal{K}}(VCP^k(a)f^k(a) + FCP^k(a)\text{Ind}(f^k(a))) \leq CA(a) \quad \forall a \in A_{\text{pst}} \tag{3.2}$$

$$f^k(a) = D^k(a) \quad \forall a \in A_c,\ k \in \mathcal{K} \tag{3.3}$$

$$\sum_b f^k((b, a^-)) = \sum_b f^k((a^+, b)) \quad \forall a \in A,\ k \in \mathcal{K}. \tag{3.4}$$

Constraints (3.2) are the capacity constraints. Satisfying the demand for each item is imposed by Constraints (3.3). Finally, Constraints (3.4) represent the classical flow conservation constraints.

It comes that the MLS-STPC problem reduces to solving the fixed charge minimum cost multi-commodity flow problem for G, that is finding the multi-commodity flow f^* that minimizes the total cost.

4. Solving method for the MLS-STPC problem by Lagrangean relaxation

As mentioned in Section 1, the MLS-STPC problem is NP-hard. We propose in the following a Lagrangean heuristic approach based on the decomposition of the problem into a facility location and a minimum cost multi-commodity flow problem.

The Lagrangean heuristic is based on the relaxation of the capacity constraints, *i.e.* Constraints (3.2) in the MIP formulation presented in Section 3. The Lagrangean multiplier vector is an indexed vector λ on the set A_{pst} of positive values. The Lagrangean function $L(f, \lambda)$ will thus be defined as follows:

$$\begin{aligned} L(f,\lambda) = & \sum_{k\in\mathcal{K}, a\in A_p} FC^k(a)\text{Ind}(f^k(a)) \\ & + \sum_{k\in\mathcal{K}, a\in A_{\text{pst}}} VC^k(a)f^k(a) \\ & + \sum_{a\in A_{\text{pst}}} \lambda(a) \sum_k FCP^k(a)\text{Ind}(f^k(a)) \\ & + \sum_{a\in A_{\text{pst}}} \lambda(a) \sum_k VCP^k(a)f^k(a) \\ & - \sum_{a\in A_{\text{pst}}} \lambda(a)CA(a). \end{aligned}$$

Therefore, the Lagrangean problem consists in $\min_{\lambda,f} L(f, \lambda)$ such that Constraints (3.3) and (3.4) are satisfied.

https://doi.org/10.1051/ro/2013050 Published online by Cambridge University Press

Let us suppose that we are provided with a Lagrangean multiplier vector $\lambda = (\lambda(a), a \in A_{\text{pst}}) \geq 0$ then, for every pair of nodes (v, w) in $V\backslash\{0, T+1\}$, we may set:

- $P^k_\lambda(v) = FC^k((0,v)) + \lambda((0,v))FCP^k((0,v))$;
- $Q^k_\lambda(v,w) = VC^k((0,v)) + \lambda((0,v))VCP^k(0,v) + Dist^k_\lambda(v,w))d^k_{w,T+1}$;
  where $\text{Dist}^k_\lambda(v,w)$ is the length of a shortest path from v to w in the network G, computed while considering every arc $a \in A_{st}$ with length $VC^k(a) + \lambda(a)VCP^k(a)$.

Before stating a property that will be useful for solving the problem, we recall briefly the definition of the standard (uncapacitated) facility location problem that will be useful to solve the MLS-STPC problem (*cf.* [10–12]). Given a (location) set of nodes X, a function g from X to the set of the non negative rational numbers, and a function l from $X \times X$ to the set of the non negative rational numbers, such that $l(x,x) = 0$ for any x in X. The facility location problem aims at finding a subset Y of X which minimizes the quantity: $\sum_{x\in Y} g(x) + \sum_{x\in X} \min_{y\in Y} l(x,y)$.

We get the following property:

Theorem 4.1. *Minimizing $L(f,\lambda)$ on the set of multi-commodity flows f which satisfy the demand constraints (3.3), reduces to solving, for every k in K, an instance of a facility location problem denoted by FL^k_λ on $X = V\backslash\{0, T+1\}$ with $g = P^k_\lambda$ and $l = Q^k_\lambda$. The related optimal value $\min_f L(f,\lambda)$ is then equal to $W^k_\lambda - \sum_{a\in A_{\text{pst}}} \lambda(a)CA(a)$ where W^k_λ is the optimal value of the facility location instance FL^k_λ.*

Proof. Solving optimally an instance of FL^k_λ provides a set of nodes Y^* that minimizes $\sum_v P^k_\lambda(v) + \sum_v \min_w Q^k_\lambda(v,w)$. Moreover, by construction, each node $v \in Y^*$ will be related to a production arc $(0,v)$ and the demand of the customer arc $(w, T+1)$ will be satisfied by using the shortest path between v and w, *i.e.* the one that minimizes the total cost $VC^k(a) + \lambda(a)VCP^k(a)$ over all the arcs of the path. Moreover, the value $\sum_{v\in Y} P^k_\lambda(v)$ is minimized, which corresponds to the minimization of the total fixed cost components in the Lagrangean function, *i.e.* $FC^k((0,v)) + \lambda((0,v))FCP^k((0,v))$. If we fix all the active production arcs as the set of arcs $(0,v)$ such that $v \in Y^*$, then every customer node w is going to get the whole demand $d^k_{w,T+1}$ from the node v in Y^* that induces the best total cost. □

We describe in the following a Lagrangean based heuristic for the MLS-STPC problem which requires solving instances of the facility location problem. We introduce the following notations that will be useful for the description of the solving approach. We will assume that a solving method called FACLOC, that uses parameter N which is the replication number, will be used to solve the instances of the facility location problem. The FACLOC solving methods used will be specified in Section 5. Two alternative algorithms will be considered. The first one will consist in using a standard MIP solver CPLEX. The second one will be based on a heuristic described in Section 5. Computing the maximal value $\max_{\lambda\geq 0} \min_f L(f,\lambda)$, may be done according to the following procedure LAG-LOT-SIZING:

https://doi.org/10.1051/ro/2013050 Published online by Cambridge University Press

LAG-LOT-SIZING(N: Integer)
$\lambda \leftarrow 0$; $W \leftarrow 0$; $\lambda_{\max} \leftarrow 0$; $n \leftarrow 0$;
While (W is improved in less than N iterations) do
Compute W_λ by applying FACLOC procedure to FL^k_λ;
If $W_\lambda > W$ then $W \leftarrow W_\lambda$; $\lambda_{\max} \leftarrow \lambda$; Update A^λ_p;
Update λ so that the sequence (λ_n) satisfies convergence conditions of the sub-gradient method ($n \to +\infty$, $\lambda_n \to 0$ and $\sum_i \lambda_i \to +\infty$);
Update the $Dist^k_\lambda$ matrices;
$n \leftarrow n + 1$;
EndWhile

The set A^λ_p corresponds to the active production arcs related to the final solution. Several formulas can be used to compute the sequence (λ_n) [14], which allow efficient updating of the Lagrangean multipliers.

LAG-LOT-SIZING ends while yielding some value W, together with some multiplier vector $\lambda_{\max}$ and a subset A^λ_p of active production arcs, *i.e.* the arcs with a strictly positive flow. However, since the capacity constraints are relaxed, the feasibility of the solution is not guaranteed. Therefore, a heuristic called PROJECTION is proposed in the following, to try to get a feasible solution.

The PROJECTION heuristic is based on solving a relaxation of the MLS-STPC problem based on the solution obtained with LAG-LOT-SIZING algorithm. The main steps of the PROJECTION algorithm are the following:

1. Solve a classical minimum cost multi-flow problem for $G' = (V', A')$ where $A' = A^\lambda_p \cup A_{\text{stc}} \cup \{(T+1, 0)\}$ and S' is the subset of S induced by A' and where all the fixed parameters are set to 0. The solution obtained at this step is denoted by $\bar{f}^k(a)$.
2. Solve the following instance of the MLS-STPC problem using LAG-LOT-SIZING for $G = (V, A)$:
   - for each arc $a \in A_c$, $D^k(a) \leftarrow D^k(a) - \bar{f}^k(a)$ (residual demands);
   - for each arc $a \in A^\lambda_p$; $CA(a) \leftarrow CA(a) - \sum_{k\in\mathcal{K}} VCP^k(a)\bar{f}^k(a) - \sum_{k\in\mathcal{K}} FCP^k(a)$ (residual capacities);
   - for each arc $a \in A_{st}$; $CA(a) \leftarrow CA(a) - \sum_{k\in\mathcal{K}} VCP^k(a)\bar{f}^k(a)$ (residual capacities).

The first step of the PROJECTION algorithm consists in solving a classical minimum cost multi-flow problem with a standard linear programming solver. The second step of the PROJECTION algorithm will provide a set of active production arcs denoted by $\bar{A}^\lambda_p$. The drawback of the PROJECTION procedure is that the obtained solution is not guaranteed to be feasible, therefore we propose to iterate the procedure in a solving approach called MLS-STPC-ALG that aims at producing a feasible solution for the problem, *i.e.* a solution that meets the demand of the customers while respecting all the capacity restrictions. The solving method is described in the following:

MLS-STPC-ALG
Stop $\leftarrow 0$; $A^\lambda_p, \bar{A}^\lambda_p \leftarrow \emptyset$;

https://doi.org/10.1051/ro/2013050 Published online by Cambridge University Press

Initialize N, Initialize an MLS-STPC instance I_G with the original parameters of the problem;
While Non Stop do
Apply LAG-LOT-SIZING(N) on instance I_G; Update A_p^λ;
If $A_p^\lambda = \emptyset$ then Stop $\leftarrow 0$ (no feasible solution);
Else
Compute $\bar{f}$ using PROJECTION given A_p^λ;
If $\bar{f}$ is null then Stop $\leftarrow 0$ (no feasible solution);
Else
Update I_G with residual demands and capacities;
If $(D^k(a) = 0)$ for all $a \in A_c$ then Stop $\leftarrow 0$ (feasible solution);
EndIf
EndIf
EndWhile

The MLS-STPC-ALG procedure terminates providing at least a lower bound on the objective value of the MLS-STPC problem. When the constraints are satisfied, a feasible solution is obtained.

5. Experimental analysis

We performed experiments on a PC AMD opteron 2.1 GHz. The MIP solver used to solve the minimum cost multi-flow problem is CPLEX. Facility location instances were alternatively handled through CPLEX and for large instances through FACLOC heuristics. We tried several instance packages based on the ones described in [9]. The generated instances correspond to the following parameters where UD means Uniform Distribution:

- number of plants: $I = 2, 3, 4, 5, 8, 10$;
- number of items: $K = 2, 3, 5, 10, 15$;
- number of periods: $T = 3, 5, 6, 10, 15$;
- variable production cost is generated using UD[0,8];
- unitary inventory (storage) cost is generated using UD[0, 2];
- variable transfer cost is generated using UD[0, 2];
- fixed setup cost is generated using UD[5, 15];
- demand is generated using UD[0, 50].

The size of an instance is defined by the triplet (I, K, T).

Capacities were generated as follows, a first flow f was randomly generated, in such a way it satisfies the demand constraints. A value of a covering parameter α was chosen in UD[0, 100], and the capacity values for the arcs were computed in order to make possible an increase of f on the equilibrium arc by $\alpha\%$.

We denote by $R_{CP}(k)$ the ratio used to measure the capacity tightness of a production plant of item k. We used the same capacity tightness average for all the instances but not the same distribution on all the couples (i, t). First, we generated

https://doi.org/10.1051/ro/2013050 Published online by Cambridge University Press

instances with $R_{CP}(k) = 30\%, k \in K$ for the experiments of Sections 5.1 and 5.2. The instances generated in Section 5.3 are such that $R_{CP}(k) = 0$, which means that the production capacity distribution is nearly the same for all the producer plants.

We handled the facility location instances in an exact way, using CPLEX, Table 1 describes the obtained gaps in percentage. The following notations are used:

- Ite: number of iterations of the MLS-STPC-ALG procedure;
- Gap: given by $\frac{W-W_{\text{Opt}}}{W_{\text{Opt}}} * 100$ which is the gap between the value W produced by MLS-STPC-ALG and the optimal value W_{Opt} obtained using the CPLEX library;
- GLg: given by $\frac{W-W_{\text{Lag}}}{W_{\text{Lag}}} * 100$ which is the gap between W and the value W_{Lag} produced by the LAG-LOT-SIZING algorithm;
- GLP: given by $\frac{W-W_{\text{LP}}}{W_{\text{LP}}} * 100$ which is the gap between W and W_{LP} produced by the LP relaxation of the MLS-STPC problem;
- GPr: given by $\frac{W_{\text{opt}}-W_{\text{opt1}}}{W_{\text{opt1}}} * 100$ which is the gap between W_{opt} and the optimal value W_{opt1} of the problem obtained by only taking into account production capacity constraints;
- GFr: given by $\frac{W_{\text{opt}}-W_{\text{opt·free}}}{W_{\text{opt·free}}} * 100$ which is the gap between W_{opt} and $W_{\text{opt·free}}$ related to the uncapacitated version of the problem;
- Def: percentage of the whole demand which could not be satisfied after the 1st iteration of the MLS-STPC-ALG algorithm.

5.1. Experiments with facility location handled through CPLEX library – part 1

Table 1 reports the results obtained when solving the facility location instances in an exact way (through CPLEX). Lagrangean relaxation yields good lower bounds. Indeed, if we compare the Gap values with the ones obtained with the LP relaxation, *i.e.* GLP, we can see that the lower bounds are close to the optimal values. The mean Gap value is equal to 0,68% while its value is 17,41% for the GLP mean value. The projection mechanism, though it usually fails in meeting the demands at the end of the first iteration, still gets close to those demands and usually meets them fast.

5.2. Experiments with facility location handled through FACLOC heuristic

When large size instances are considered, heuristic procedures are used to handle the facility location problem. It is interesting to study in such a case, the way the solution induced by the application of such a heuristic procedure is going to impact the final result. In order to understand it, we used the instances described in Table 1, and, for every instance, we computed the mean error (in percentage) performed by the FACLOC procedure used, and compared it with the final error. The FACLOC like heuristic which we used is based on designing a randomized GRASP

https://doi.org/10.1051/ro/2013050 Published online by Cambridge University Press

TABLE 1. Gap analysis.

Size	Ite	Gap (%)	GLg (%)	GLP (%)	GPr (%)	GFr (%)	Def (%)
(3,5,3)	1	0	5.4	15.6	0.8	85	0
(3,5,3)	1	0	7.5	12.7	0.6	54	0
(3,5,3)	1	0	0.8	15.9	5.8	32.5	0
(4,10,6)	1	1.7	0.2	15.6	2.9	21.5	0
(4,10,6)	1	0	0.2	20.3	3.8	9.2	0
(4,10,6)	2	1.0	5.4	13.8	20.0	25.4	3.5
(5,10,10)	1	0.7	0.6	19.7	5.2	20.2	0
(5,10,10)	1	0.7	1.5	15.3	6.7	19.1	0
(5,10,10)	1	0.1	0.4	20.2	5.5	14.7	0
(8,10,15)	1	0	0.9	13.5	2.6	23.6	0
(8,10,15)	1	0.8	3.4	12.8	3.7	49.0	0
(8,10,15)	2	1.5	4.5	19.4	7.5	58.3	2.2
(10,15,15)	1	1.2	2.6	23.7	3.4	37.2	0
(10,15,15)	1	0	0.9	14.5	3.5	29.6	0
(10,15,15)	3	2.5	6.5	28.2	11.8	82.0	4.8

greedy and descent scheme to construct a solution. An initial solution is obtained by considering that each demand for a plant related to a given customer is produced at the same plant. A shortest path computation between plants follows and allows to transfer some demands between plants, thus leading to the improvement of the initial solution. It can be interesting to use a more sophisticated algorithm such as the one described in [13].

The following parameters are used:

- Ite-Approx (resp. Def-Approx) represents parameter Ite (resp. Def) defined in Section 5.1, for the case where the facility location problem is handled through a heuristic way;
- Gap-Approx and GLg-Approx are the increase rate (in %) with relation respectively with Gap and GLg of the experiments conducted in Section 5.1, for the case where the facility location problem is handled through a heuristic way;
- Gap-Loc-Approx is the mean error (in %) induced by the use of the FACLOC heuristic instead of a MIP solver when solving the facility location sub-problem.

A sensibility analysis is conducted. Table 2 provides the obtained results.

We can see that GLg-Approx increase closely follows the gap between the optimal solution and the result produced by the heuristic. Still, impact on the final Gap value was more difficult to predict, due to the dependency of the behaviour of the projection mechanism on the characteristics of the arc production set obtained at the end of the LAG-LOT-SIZING procedure.

https://doi.org/10.1051/ro/2013050 Published online by Cambridge University Press

TABLE 2. Sensibility analysis: heuristic vs. exact method for FACLOC.

Size	Ite-Approx	Gap-Approx (%)	GLg-Approx (%)	Gap-Loc-Approx (%)	Def-Approx (%)
(3,5,3)	2	10.8	6.7	6.5	2.3
(3,5,3)	1	2.0	2.0	2.1	0
(3,5,3)	1	12.6	12.8	12.5	0
(4,10,6)	2	9.9	5.1	3.8	1.8
(4,10,6)	1	6.8	6.7	6.4	0
(4,10,6)	1	0.1	1.5	1.9	0
(5,10,10)	2	11.0	9.6	8.5	5.2
(5,10,10)	2	20.7	15.0	13.4	2.1
(5,10,10)	1	8.4	6.9	7.7	0
(8,10,15)	1	5.0	5.1	4.8	0
(8,10,15)	1	14.5	16.4	15.2	0
(8,10,15)	3	10.2	4.5	3.9	6.2
(10,15,15)	1	12.5	14.9	10.6	0
(10,15,15)	1	7.0	6.8	7.3	0
(10,15,15)	2	3.2	7.6	5.5	2.8

TABLE 3. Gap analysis.

Size	Ite	Gap (in %)	GLg (in %)	GLP (in %)	GFr (in %)
(3.5.6)	2	2.01	4.90	3.54	52.16
(3.5.6)	2	1.29	3.94	3.93	30.16
(3.5.6)	2	0.85	6.45	2.28	34.47
(3.5.6)	2	1.42	3.70	3.70	48.22
(3.5.6)	2	0.00	10.51	1.28	62.46
(3.3.10)	1	2.13	5.76	6.50	57.02
(3.3.10)	2	0.00	6.56	5.08	62.74
(3.3.10)	2	0.98	6.37	6.29	94.34
(3.3.10)	2	1.69	18.57	6.22	112.65
(3.3.10)	2	0.11	8.04	3.63	65.13
(5.5.15)	2	4.40	6.41	18.16	16.71
(5.5.15)	2	2.71	4.25	12.34	18.49
(5.5.15)	2	5.04	6.20	18.88	16.71
(5.5.15)	2	4.38	5.69	15.35	20.19
(5.5.15)	2	2.44	3.71	13.00	20.63

5.3. EXPERIMENTS WITH FACILITY LOCATION HANDLED THROUGH CPLEX LIBRARIES – PART 2

As shown in Section 5.1, we solved the facility location problem in exact way but we generated other kind of instances with $R_{cp}(k) = 0, k \in K$ and other sizes. Table 3 reports the results of three sets of five instances.

In Table 3, we can notice that even if the capacity repartition seems to lead to more difficult instances, the proposed procedure solves efficiently the problem.

https://doi.org/10.1051/ro/2013050 Published online by Cambridge University Press

We can notice that the optimal solution is reached for some instances. However, the gap values increase in comparison to the results observed in Table 1, but are at most equal to 5,04%. The impact of the capacity constraints can also be measured by the high increase of the GFr gap values.

The CPU times for instances of size $(5, 10, 10)$ are less than one minute in average for the FACLOC heuristic while they reach 10 min for instances of size $(10, 15, 15)$. The CPU times of the FACLOC heuristic greatly depend on the parameters of the method such as the number of iterations of the GRASP heuristic. The CPU times reach 60 min in average when using CPLEX for instances of size $(10, 15, 15)$.

A more detailed analysis on the computation times should be conducted on more dedicated heuristic and specialized flow algorithms. The contribution of the paper here was to test if the decomposition principle for solving the problem was accurate and could lead to good solutions. It should be pursued by tuning fast solving methods to tackle large instances of the problem.

6. Conclusion

In this paper, we address an NP-hard multi-item multi-plant lot-sizing problem with production, transfer and storage capacity constraints. This work generalizes the problem considered by Sambivasan and Yahya [9]. We described the way the problem might be decomposed, into facility location and network multi-commodity flow problems. Since both problems have been widely studied, casting lot-sizing problems into such a decomposition framework opens the way to design generic software components. An efficient Lagrangean heuristic is proposed. The maximum deviation from the optimal value does not exceed 5% when the facility location is solved to optimality for all the instances of the test-bed. Several research perspectives arose from this study. We can extend the model by considering fixed parameters for the storage and the transfer aspects. We can also allow some latencies on the transfer operations. Moreover, specific algorithms should be used for FACLOC procedure instead of a MIP solver. This is also the case for the minimum cost multi-flow problem. Other subgradient techniques could also be used to speed up the Lagrangean algorithm. Finally, the good quality of the solutions suggests the implementation of a branch-and-bound algorithm to find optimal solutions to the problem.

Acknowledgements. This work has been partially funded by ANR under research contract LMCO/UPMC number C076095.

References

[1] I. Barany, T.J. van Roy and L.A. Wolsey. Strong formulations for multi-item capacitated lot-sizing. *Manag. Sci.* **30** (1984) 1255–1261.

[2] G. Bitran and H. Yanasse, Computational complexity of the capacitated lot size problem. *Manag. Sci.* **28** (1982) 1174–1186.

https://doi.org/10.1051/ro/2013050 Published online by Cambridge University Press

[3] W. Chen and J. Thizy, Analysis of relaxations for the multi-item capacitated lot-sizing problem. *Annal. Oper. Res.* **26** (1990) 29–72.
[4] M. Florian and M. Klein, Deterministic production planning with concave costs and capacity constraints. *Manag. Sci.* **18** (1971) 12–20.
[5] M. Florian, J. Lenstra and A.R. Kan, Deterministic production planning: algorithms and complexity. *Manag. Sci.* **26** (1980) 669–679.
[6] R. Jans and Z. Degraeve, Modeling industrial lot sizing problems: a review. *Int. J. Prod. Res.* **46** (2008) 1619–1643.
[7] J. Maes and L. van Wassenhove, Multi-item single-level capacitated lot-sizing heuristics: a general review. *J. Oper. Res. Soc.* **11** (1988) 991–1004.
[8] M.C.V. Nascimento, M.G.C. Resende and F.M.B. Toledo, Grasp heuristic with path-relinking for the multi-plant capacitated lot sizing problem. *Eur. J. Oper. Res.* **200** (2010) 1–17.
[9] M. Sambivasan and S. Yahya, A Lagrangean based heuristic for multi-plant, multi-item, multi-period capacited lot sizing problems with inter-plant transfers. *Comput. Oper. Res.* **32** (2005) 537–552.
[10] G. Ghiani, F. Guerrero and F. Musmanno, The capacitated plant location problem with multiple facilities in the same site. *Comput. Oper. Res.* **29-13** (2002) 1903–12.
[11] G. Cornuejols, G.L. Nemhauser and L.A. Wolsey, The uncapacited facility location problem. In *Discrete location theory*, edited by B. Mirchandani, R.L. Francis (1990) 119–171.
[12] B. Mirchandani and R.L. Francis, *Discrete location theory*. Wiley, J. Sons (1990).
[13] M.G. Resende and R.F. Werneck, A hybrid heuristic for the p-median problem. *J. Heuristics* **10** (2004) 59–88.
[14] M. Minoux, *Programmation Mathématique*, Tome 1. Dunod Ed (1983).
[15] S. Wu and H. Golbasi, Multi-Item, Multi-Facility Supply Chain Planning: Models, Complexities, and Algorithms. *Comput. Optim. Appl.* **28** (2004) 325–356.

https://doi.org/10.1051/ro/2013050 Published online by Cambridge University Press