Integrating Side Payments into Collaborative Planning for the Distributed Multi-Level Unconstrained Lot Sizing Problem

Jörg Homberger*, Hermann Gehring† and Tobias Buer‡

*University of Applied Sciences Stuttgart, Germany
joerg.homberger@hft-stuttgart.de

†Information Systems Research Group, University of Hagen, Germany
hermann.gehring@fernuni-hagen.de

‡Computational Logistics, Cooperative Junior Research Group of University of Bremen and ISL - Institute of Shipping Economics and Logistics, Germany
tobias.buer@uni-bremen.de

Abstract

Collaborative planning mechanisms coordinate the decisions of multiple, autonomous, and self-interested decision makers under asymmetric information. The approach proposed in this paper extends collaborative planning for the distributed multi-level uncapacitated lot-sizing problem by integrating compensation payments. Compensation or side payments provide an incentive for individual decision makers to accept inferior local solutions that may direct the search to superior global solutions for a coalition of decision makers. The approach uses neighborhood search, voting-based solution acceptance criteria and takes into account varying side payments which are negotiated. Based on 272 benchmark instances the computational study shows that the presented approach is able to achieve substantial progress compared to earlier methods. It therefore is beneficial to incorporate side payments into negotiation processes based on collaborative search.

1. Introduction

Methods for collaborative planning support a coalition of self-interested decision makers (briefly, agents) coordinating their individual plans [1]. Coordinated planning is important when a global optimum cannot be achieved if the agents pursue their local plans independently from each other. In that case, a coordinated solution generated by collaborative planning leads to added benefits. These added benefits can be allocated to the agents such that each agent is better off compared to the non-coordinated case.

One application area of collaborative planning is production planning in a supply chain of independent companies, for example. The companies in a supply chain are connected via physical, financial, and information flows. A shared greater goal of the companies in such a supply chain is to fulfill the (forecasted) demand of their final customers. A traditional approach to coordinate production planning is hierarchical planning [32]. A central decision maker, who usually represents the original equipment manufacturer, plans the production and storage of items for all members of the supply chain. A representative model for this situation is the well-known multi-level lot-sizing problem. Because this model features several characteristics of real-world production problems we use it as the basic operations planning problem in this paper. However, using hierarchical planning to solve a multi-level lot-sizing problem neglects the existence of private information of the companies, conflicting goals of the companies, and the missing authority of the central decision maker to enforce decisions for all members of the supply chain. To overcome these deficits, collaborative planning is discussed [5].

Collaborative planning scenarios are characterized by asymmetric information and conflicting objective criteria [2]. That is, the agents rely on the same resources and each agent wants to optimize his or her local objective function. At this, an agent’s sensitive planning data is considered private information. In production planning this may be, e.g., the backlog of orders, the workload of machines, or their unit costs. Preferably, the agents aim to share as few
private information as possible or none at all. For these reasons, it is not feasible to use central planning via an omniscient decision maker that computes a global optimal plan. Rather, methods for collaborative (or decentralized, which is used synonymous here) planning are required which coordinate plans. As long as the agents’ autonomy is preserved, they have an incentive to cooperate because the added benefits through the coordinated plan overcompensates them for deviating from their local optimal plans.

After a coordinated plan is computed, those agents that are better off may compensate those agents that are inferior. In a business situation, the transfer of utility is usually made via side payments. Side payments (over-)compensate agents for disadvantages suffered from deviations to their local plans generated during isolated planning. Because side payments are possible, coordination of planning between agents is reasonable. However, in most collaborative planning approaches (see Section 2) the agents are unaware of future side payments linked to a proposed plan. This implies serious drawbacks for the search of a jointly agreed plan because the information about additional side payments clearly influences an agent’s decision whether to accept or reject a new proposal for a plan. Instead of computing the side payments ex post, we propose an approach that computes and negotiates side payments ex ante, i.e., before a solution is finally accepted by all agents. This happens during the actual search for a coordinated solution.

The goal of this paper is to present and to evaluate a way of integrating the negotiation of side payments between agents into collaborative planning based on neighborhood search. This is done using the distributed multi-level uncapacitated lot-sizing problem (DMLULSP) as an example. The proposed collaborative local search with side payments (CSP) includes two phases. Phase I of CSP uses a modified variant of collaborative local search with complete approval voting [3]. The side payments are integrated into Phase II. They are negotiated between the agents. Negotiation is implemented via voting-based criteria that determine if a generated solution is (either temporarily or ultimately) accepted as a new best solution on which further moves are performed. As is shown below, these concepts contribute significantly to find superior solutions.

The paper is structured as follows. Section 2 briefly reviews the literature. Section 3 describes the distributed multi-level lot-sizing problem at hand. Section 4 introduces the two-phase collaborative neighborhood search (CSP) which also negotiates side payments between the agents to solve the DMLULSP. The performance of CSP is evaluated in Section 5 by means of a computational study based on 272 benchmark instances. Section 6 concludes the paper.

2. Review of the literature

The state of the art in collaborative planning is discussed in [4], [5]. Overviews of automated negotiation approaches relevant to collaborative planning are given by [6], [7]. Applications of collaborative planning to supply chains are discussed, e.g., by [8]–[10].

Lot-sizing problems are of high relevance in modern supply chains [11]. Surveys of the rich literature on lot-sizing problems of a single decision maker are given for example by [12]–[14]. Metaheuristic solution approaches for variants of multi-level lot-sizing problems of a single decision maker presented by, e.g., [15]–[24]. Those approaches provide a natural extension point for the development of collaborative metaheuristics for multi-level lot-sizing.

In this paper, the presented collaborative planning approach is applied to a distributed variant of the well-known multi-level uncapacitated lot-sizing problem (MLULSP) introduced by [25]. The distributed MLULSP (DMLULSP) was presented by [26] and assumes several local and selfish agents with private information instead of a single agent with full information. The existing collaborative solution approaches for the DMLULSP differ by their used metaheuristic search principles and the techniques used to aggregate the agents’ preferences ([26], [27] use simulated annealing, [28] uses an evolutionary strategy, and [3], [29], [30] use ant colony optimization). The DMLULSP covers some important features of real world problems. There are several final products, a multi-level production structure, and a trade-off between inventory and setup costs. Coordination is difficult because there are agents with private information and conflicting objectives. Finally, the problem is computationally challenging, as it is NP-hard for general product structures [31]. As to computational experiments, a variety of benchmark instances is available in the literature which enable comparative tests including other approaches. In [32], a multi-level capacitated lot-sizing problem is studied. The additional capacity constraints are even closer to real world requirements, however, the negotiation approach is designed for coordination between two agents while the approach presented here is applicable to any number of agents.

In addition to the DMLULSP, the literature discusses another variant of distributed lot-sizing. In the approach by [33], [34], auctions are used instead of voting mechanisms to coordinate planning. The ap-
Table 1. Notation for the DMLULSP.

<table>
<thead>
<tr>
<th>Problem parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$ number of items</td>
<td></td>
</tr>
<tr>
<td>$n$ number of production periods</td>
<td></td>
</tr>
<tr>
<td>$I$ set of items, $I = {1, \ldots, m}$</td>
<td></td>
</tr>
<tr>
<td>$T$ set of possible production periods, $T = {1, \ldots, n}$</td>
<td></td>
</tr>
<tr>
<td>$t_i$ inventory holding costs per period and per unit of item $i \in I$</td>
<td></td>
</tr>
<tr>
<td>$t_s$ lead time required to assemble, manufacture, or purchase item $i \in I$</td>
<td></td>
</tr>
<tr>
<td>$r_{ij}$ number of items $i$ required to produce one unit of item $j$ with $i, j \in I$, $i \neq j$</td>
<td></td>
</tr>
<tr>
<td>$\Gamma^+(i)$ set of all direct successors of item $i \in I$, $\Gamma^+(i) \subset I$</td>
<td></td>
</tr>
<tr>
<td>$\Gamma^-(i)$ set of all direct predecessors of item $i \in I$, $\Gamma^-(i) \subset I$</td>
<td></td>
</tr>
<tr>
<td>$d_{it}$ exogenous demand (unit of quantity) for item $i \in {j \in I</td>
<td>\Gamma^+(j) = \emptyset}$ in period $t \in T$</td>
</tr>
<tr>
<td>$M_{it}$ a sufficiently large number, $i \in I$, $t \in T$, e.g. $M_{it} = \sum_{t' \geq t} d_{it'}$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision variables</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{it}$ endogeneous demand (unit of quantity) of item $i \in {j \in I</td>
<td>\Gamma^+(j) \neq \emptyset}$ in period $t \in T$</td>
</tr>
<tr>
<td>$l_{it}$ inventory (unit of quantity) of item $i \in I$ at the end of period $t \in T$</td>
<td></td>
</tr>
<tr>
<td>$x_{it}$ lot-size (unit of quantity) of item $i \in I$ in period $t \in T$</td>
<td></td>
</tr>
<tr>
<td>$y_{it}$ binary setup decision, $y_{it} = 1$ if item $i \in I$ is produced in period $t \in T$ and $y_{it} = 0$ otherwise</td>
<td></td>
</tr>
</tbody>
</table>

3. A model of the distributed lot-sizing problem with side payments

3.1. The multi-level uncapacitated lot-sizing problem (MLULSP) of a single decision maker

In the style of [3], a model of the well-known multi-level uncapacitated lot-sizing problem (MLULSP, cf. [25], [38], [15]) of a single decision maker is presented first. The notation is given in Table 1. The MLULSP assumes a single decision maker who is aware of all information required for planning, especially, the setup costs and inventory holding costs per item and period. The MLULSP of a single decision maker with full information is given by the formulas (1) to (8).

\[
\min f^{nd}(y) = \sum_{i \in I} \sum_{t \in T} (s_i \cdot y_{it} + h_i \cdot l_{it}) \quad (1)
\]

s. t. \quad \begin{align*}
\quad l_{it} &= l_{i,t-1} + x_{it} - d_{it}, & \forall i \in I, \forall t \in T, (2) \\
\quad l_{i,0} &= 0, & \forall i \in I, (3) \\
\quad l_{it} &\geq 0, & \forall i \in I, \forall t \in T \setminus \{0\}, (4) \\
\quad d_{it} &= \sum_{j \in \Gamma^+(i)} r_{ij} \cdot x_{j,t+\ell_i}, & \forall i \in \{j \in I | \Gamma^+(j) \neq \emptyset\}, t \in T, (5) \\
\quad x_{it} - M_{it} \cdot y_{it} &\leq 0, & \forall i \in I, \forall t \in T, (6) \\
\quad x_{it} &\geq 0, & \forall i \in I, \forall t \in T, (7) \\
\quad y_{it} &\in \{0, 1\}, & \forall i \in I, \forall t \in T. (8)
\end{align*}

The objective function (1) of the MLULSP minimizes the total costs $f^{nd}$ of the central decision maker. These are expressed by the objective function (1) which sums up the setup costs and the stockholding costs for all items $i \in I$ over all periods $t \in T$. The inventory balance is guaranteed by (2). For all items, the inventory of the first period $t = 0$ is zero (3) and for remaining periods non-negative (4). For each period, the demands $d_{it}$ for the final products $i \in I$ with $\Gamma^+(i) = \emptyset$ are given. The demands for the remaining items are determined by (5). These constraints ensure that the production of item $j$ in period $t + \ell_i$ triggers a corresponding demand $d_{it}$ for all $i \in \Gamma^-(j)$, that is, a demand for each item $i$ preceding item $j$ in the bill of materials. Without loss of generality, $r_{ij} = 1$ is assumed in (5). The lot-size $x_{it}$ is non-negative (7). If $x_{it} > 0$, that is, item $i$ is produced in period $t$, then $y_{it} = 1$, otherwise $y_{it} = 0$. This is enforced by the constraints (6) and (8).
The MLULSP is NP-hard for general gozintograph structures [31], i.e., for hierarchical, multi-level product structures where raw materials (Γ−(i) = ∅) and intermediate products (|Γ−(i)|, |Γ+(i)| > 0) may be required as pre-products by multiple intermediate or final products.

3.2. Distributed lot-sizing with side payments as a group decision problem

In contrast to the MLULSP the distributed multi-level uncapacitated lot-sizing problem (DMLULSP) introduced in [26] assumes a set A of decision makers (or agents) that jointly solve the MLULSP. An agent \(a \in A\) might represent the decision maker of a profit center or an independent company and the agents might have to interact in a supply chain.

All agents in \(A\) are jointly responsible to produce all items in \(I\). The production responsibility is assigned as follows: Each agent \(a \in A\) is responsible to produce the set of items \(I_a\) with \(\cup_{a \in A} I_a = I\) and \(\cap_{a \in A} I_a = \emptyset\).

Each agent is autonomous and self-interested. Therefore, the individual objective function \(f_a\) of agent \(a (a \in A)\) is to minimize his or her total local costs for producing the items \(I_a\), that is,

\[
\min f_a(y) = \sum_{i \in I_a} \sum_{t \in T} (s_i \cdot y_{it} + h_t \cdot l_{it}). \tag{9}
\]

In order to take the optional side payments \(p_a\) for an agent into account, an alternative local cost function (9) of an agent \(a\) is also used:

\[
\min f'_a(y) = f_a(y) - p_a. \tag{10}
\]

Positive side payments \(p_a > 0\) of an agent \(a \in A\) indicate the amount of money \(a\) pays to one or more agents in \(A \setminus \{a\}\). Negative side payments \(p_a < 0\) represent the amount of money agent \(a\) receives from all other agents \(A \setminus \{a\}\). The side payments of all agents sum up to zero.

\[
\sum_{a \in A} p_a = 0, \tag{11}
\]

\[
p_a \in \mathbb{R}, \quad \forall a \in A. \tag{12}
\]

Finally, the DMLULSP assumes asymmetric information regarding the cost parameters \(s_i\) and \(h_t\). That is, for all items \(i \in I_a\) produced by an agent \(a (a \in A)\), only agent \(a\) knows the values of \(s_i\) and \(h_t\); all other agents \(a' (a' \in A, a' \neq a)\) are not aware of the values of \(s_i\) and \(h_t\) \((i \in I_a)\). Agent \(a\) does not want to reveal these to other agents during (collaborative) planning, therefore these cost parameters are denoted as private information of agent \(a\). If this information would be common knowledge, price negotiations between agents might be negatively affected. On the other hand, symmetric information or public information, i.e., information available to all agents, is assumed regarding the bill of materials. This assumption can be justified by some kind of common industry knowledge or joint development of products (e.g., collaborative engineering). With respect to the side payments \(p_a (a \in A)\), we assume the agents disclose this information to the neutral mediator but not the other agents.

The DMLULSP consists of the constraints (2) to (8) and the objective function (13) which minimizes the total global costs:

\[
\min f(y) = \sum_{a \in A} f_a(y). \tag{13}
\]

The DMLULSP with side payments consists of the constraints (2) to (8), (11), (12) and the objective function (14) which minimizes the total global costs:

\[
\min f'(y) = \sum_{a \in A} f'_a(y). \tag{14}
\]

Note, \(f(y) = f'(y)\) and each solution \(y\) that is feasible under the constraints of the DMLULSP is also feasible under the constraints of the DMLULSP with side payments and vice versa.

In comparable scenarios, the total global costs are also considered of interest by [1], [9], [32], [39]. In the following, the functions \(f\) (13) and \(f'\) (14) are simply referred to as total cost function or global cost function. Function \(f_a\) (9) is also denoted as individual or local cost function of agent \(a \in A\). Under many conditions, the minimization of the local cost functions conflicts with the objective of minimizing the global cost function.

4. A two-phase collaborative local search approach with side payments (CSP)

4.1. Solution encoding and main parameters

The presented approach is denoted as collaborative local search with side payments (CSP). CSP is based on encoded solutions. It is well-known, that for the MLULSP the decision on the optimal lot sizes \(x\) may be derived from the optimal setup decisions \(y\), see [15]. Therefore, the heuristic search need not contemplate on the actual lot sizes but rather focus on finding optimal setup decisions. Instead of modifying the binary setup
decision variables $y$ during the search, the search uses an encoded solution $e$. The encoding has been introduced in [3] and works as follows.

$$(e_{it} = 0) \iff (y_{it} = 0),$$

$$(e_{it} = 1) \iff (y_{it} = 0 \lor y_{it} = 1),$$ \hspace{1cm} (15)

A value of $e_{it} = 0$ defines, that item $i$ may not produced in period $t$. However, a value of $e_{it} = 1$ indicates, that a production of item $i$ in period $t$ is possible, but not mandatory. The encoded solution contributes to keep at least some information private, because actual setup decisions are not published. Furthermore, production is always allowed in the first period, i.e., $e_{i1} = 1$ is fixed for all items $(i \in I)$ and not changed during the search. For this reason, an encoded solution can always be decoded to a feasible solution because the total demand for an item $i \in I$ for the planning horizon may be produced in period $t = 1$ due to lacking capacity constraints in the DMLULSP. Therefore, the search spends almost no resources on obtaining and preserving feasibility of solutions, even under information asymmetry.

In addition to using encoded solution, the search process requires three main parameters. The number $r$ of negotiation rounds which is used as termination criterion. The total rounds are divided into the number $r^I$ and $r^{II}$ of negotiation rounds in Phase I and Phase II of the heuristic $(r := r^I + r^{II})$. In Phase II, which includes the side payments, the initial fractional payment $\delta_a$ and a payment increment $\Delta_a$ become relevant. $\Delta_a$ is the constant amount of money by which agent $a$ increases his or her willingness to pay if a solution with lower local costs $f_a$ compared to the previously jointly agreed solution is achieved. $\Delta_a$ is initially defined as $\Delta_a := \delta_a \cdot f_a$. For the computational study, we assume $\Delta_a$ and $\delta_a$, respectively, are equal for all agents $a \in A$.

4.2. Phase I: Multiple-neighborhood hill climbing without side payments

Phase I is a hill climbing based neighborhood search using the three neighborhoods jump, bitflip, and bitswap. An overview is given by Algorithm 1. The search process is controlled by a neutral mediator, see [3], [9], [10]. The agents enter the actual negotiation in Step 4. In Step 4, the mediator successively presents each neighborhood solution to the agents which vote on accepting the new solution.

The search begins with the initial encoded solution $e_{i,t} = 1$ for $i = 1 \ldots m$ and $t = 1 \ldots n$. That is, each agent is allowed to produce each item in each period, if the agent thinks this is reasonable. Solutions are modified by local search based on three neighborhood structures and a voting-based first fit acceptance criterion. The three moves used to generate neighborhood solutions are:

- The move jump sets $e_{it} \leftarrow \neg e_{it}$ for a randomly chosen $i,t$. The cardinality of the jump-neighborhood is 1.
- The move bitflip sets $e_{it} \leftarrow \neg e_{it}$, it is performed for all $i = 1 \ldots m$ and all $t = 1 \ldots n$. The cardinality of the bitflip-neighborhood is $mn$.
- The move bitswap interchanges the values of two entries in the matrix $e$, i.e.,

  $$\text{tmp} \leftarrow e_{i_1,t_1}; \quad e_{i_1,t_1} \leftarrow e_{i_2,t_2}; \quad e_{i_2,t_2} \leftarrow \text{tmp}$$

It is performed for all pairs $i,t$ with $i = 1 \ldots m; t = 1 \ldots n$. The cardinality of the bitswap-neighborhood is $mn(m-1)(n-1)/2$.

After each bitflip-move and after each bitswap-move, solution acceptance is tested. Acceptance of a solution is performed via approval voting. A new solution $e$ is only accepted as a new mutual best solution $e^*$, if each agent $a \in A$ is able to reduce his local cost, i.e.,

$$f_a(e) \leq f_a(e^*) \quad \text{for all } a \in A. \quad (17)$$

As formula (17) shows, solution acceptance in Phase I does not depend on side payments. We denote the execution of a bitflip-move or a bitswap-move together with the following voting procedure as one round of negotiation. Phase I terminates after $r^I$ negotiation rounds.

4.3. Phase II: Side payments negotiations

Phase I terminates after $r^I$ solutions have been generated. We denote the jointly agreed solution $e^*$ at this point as local negotiation optimum. The term is
chosen, because – presume $r^I$ is sufficiently large – the search usually stagnates. This means, in all likelihood, an agent is only able to decrease his local costs if the local costs of at least another agent increase.

Phase II uses side payments to overcome such a local negotiation optimum. A side payment is an amount of money that agents with decreasing local costs (compared to their solution in the local negotiation optimum) pay to the agents with increasing local costs, such that each agent is better off. Let $p_a$ denote the payment ($p_a > 0$) of agent $a$ or rather the amount of money agent $a$ receives ($p_a < 0$).

Because the local costs of the agents are private information, it is nontrivial for the mediator to identify side payments that lead to indifference between alternative solutions. Our idea to integrate side payments is as follows. In the initial step of Phase II, each agent $a \in A$ defines an incremental value $\Delta_a$ of his willingness to pay for an improved solution. For simplicity, $\Delta_a$ is defined as a percentage $\delta_a$ of $f_a(e^*)$, i.e.,

$$\Delta_a = \delta_a \cdot f_a(e^*). \quad (18)$$

The percentage $\delta_a$ is private information of agent $a$. However, we assume $\Delta_a$ is also known to the mediator who keeps this information secret. It is not known to the other agents.

The actual search continues with the local negotiation optimum solution as initial solution $e^*$. Phase II uses the same neighborhood structures and the same sequence as Phase I (cf. Algorithm 1). However, there are now two different acceptance criteria.

The first acceptance criterion is again an approval voting procedure, which now takes into account side payments. A new solution $e$ is jointly accepted as a new best solution $e^*$, if each agent $a \in A$ is able to reduce the local cost including the necessary side payments, i.e.,

$$f_a(e) - p_a \leq f_a(e^*) \quad \text{for all } a \in A. \quad (19)$$

The second acceptance criterion is shown by Algorithm 2. It relaxes and modifies rule (19) so that it is sufficient that only $|A| - 1$ agents improve on their solution, while the remaining agent temporarily accepts an inferior solution.

Phase II distinguishes between the solutions $e$, $e^{tmp}$, and $e^*$ which denote a current solution, a temporary accepted solution, and a jointly accepted solution. A temporary accepted solution $e^{tmp}$ is used as means to facilitate exploring the search space when there is little progress of the search.

Let $R$ be a subset of agents ($R \subseteq A$). A solution $e^{tmp}$ is called temporary accepted, if

```plaintext
tmpAccept \leftarrow false
// set R of agents that reject e
R \leftarrow \{a \in A | f_a(e) > f_a(e^*)\}
if |R| = 1 then
    p^{tmp}_r \leftarrow p_r - \sum_{a \in A \setminus R} \Delta_a \quad \text{for all } R
    if f_r(e) - p^{tmp}_r \leq f_r(e^*) then
        tmpAccept \leftarrow true
        p_r \leftarrow p^{tmp}_r
    foreach a \in A \setminus R do
        p_a \leftarrow p_a + \Delta_a
    end
    e^* \leftarrow e
else
    // incr. willingness to pay
    foreach a \in A do
        \Delta_a \leftarrow \Delta_a + \delta_a f_a(e^{tmp})
    end
end
tmpAccept
return tmpAccept
```

Algorithm 2: Acceptance rule in Phase II with side payments

a) there is at most one agent $r, r \in R$ with increased local costs, i.e. $f_r(e^{tmp}) > f_r(e^*)$ and
b) $|A| - 1$ agents have lower local costs compared to $e^*$, i.e. $f_a(e^{tmp}) \leq f_a(e^*)$ for all $a \in A \setminus R$ and

c) the agent $r$ would be better off, if $r$ receives the side payments $f_r(e^{tmp}) - \sum_{a \in A \setminus R} \Delta_a \leq f_a(e^*)$.

Note, in b) the agents do not take into account the side payments $\Delta_a, a \in A \setminus R$ they have to pay to agent $r \in R$ with their temporary acceptance decision. Again, Phase II is terminated when $r^{II}$ new solutions have been generated.

5. Evaluation

5.1. Test set up and parameter setting

The goal of the evaluation is to study whether side payments implemented in the way of Section 4.3 improve collaborative planning. Our computational study is based on 272 benchmark instances which are available at http://www.dmlulsp.com. Originally, the instances have been introduced by [40], [41], and [15] for the non-distributed MLULSP. They are divided into three classes of small, medium, and large instances denoted as s, m, and l. These instances have been extended in [26] for the distributed scenario, taking into account two ($|A| = 2$) and five agents ($|A| = 5$).
Table 2. Characteristics of used benchmark instances

<table>
<thead>
<tr>
<th>class</th>
<th>group</th>
<th></th>
<th>no. of instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>s2</td>
<td>2, 5, 12</td>
<td>96</td>
</tr>
<tr>
<td>s5</td>
<td>5, 5, 12</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>m</td>
<td>m2</td>
<td>2, 40, 50, 12, 24</td>
<td>40</td>
</tr>
<tr>
<td>m5</td>
<td>5, 40, 50, 12, 24</td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>

For this, each item has been assigned to exactly one agent that produces the item. Demand and cost data of the original MLULSP instances have been adopted and were not modified. We exclude the set of large instances from the test and use only instance groups s2, s5, m2, and m5 (cf. Table 2). Preliminary tests showed that our neighborhoods are not powerful enough to compute solutions of very good quality for large instances. Nevertheless, the medium sized instances cover up to 50 items which have to be produced during a planning horizon of up to 24 periods which is still reasonable large.

The effectiveness of a decentral optimization method is measured via a comparison with the solution quality obtained in the central case (i.e., a single omniscient decision maker) as proposed in [1]. For a given instance, the gap \( G(y) \) of the computed solution \( y \) is calculated by

\[
G(y) = \frac{f(y) - f^{nd}(y^{bk})}{f^{nd}(y^{bk})} \times 100. \tag{20}
\]

The value \( f^{nd}(y^{bk}) \) indicates the objective function value of the best-known solution for the non-distributed MLULSP which are taken from the literature [15], [18], [19], [23], [26], see [3] for an aggregation.

5.2. Effects of side payments

Table 3 shows the average gap \( G \) achieved by three methods for each tested instance group. For all three methods, the number of generated solutions \( r \) per instance group are equal. Each method generated \( r = 50,000 \) solutions per instance in group s2 or s5, while \( r = 200,000 \) solutions were computed for each instance in group m2 or m5. The rightmost column indicates a reference value from the literature achieved by the method denoted as collaborative ant colony metaheuristic (CACM) introduced in [3]. Note, the approach CACM also used 50,000 and 200,000 negotiation rounds, respectively. In terms of the average gap \( G \) it may be characterized as one of the most competitive approaches for the instance groups s2, s5, m2, and m5. The column Phase I shows the average gap achieved by computing \( r \) solutions with the collaborative neighborhood search of Phase I only (see Section 4.2). Column Phase I+II shows the results when side payments are considered as a means to escape from local negotiation optima.

For each class of instances, the same parametrization of Phase I and Phase II was used, in particular \( r^I = r^{II} = 0.5r \). The sole exception is parameter \( \delta_a \). The instances of group s2, s5, and m2 were solved using \( \delta_a = 10^{-8} \) (for all \( a \in A \)) and the instances of group m5 were solved using \( \delta_a = 10^{-4} \) (for all \( a \in A \)). Parameter \( \delta_a \) influences agent’s \( a \) willingness to pay and the increment \( \Delta_a \) by which the side payment is increased when no jointly accepted solution is found. A smaller value is used to solve the instances of group m5, because the objective function values are an order of magnitude higher compared to s2 and s5. In addition, for the five agent case a single agent receives the sum of payments of four agents which maybe quite large. This can lead to situations were the agent that accepts increased local costs receives payments that are considerably higher than the suffered cost increase (see Section 5.3). In turn, Phase II converges early and is not able to escape from the local negotiation optimum.

Table 3. Average gap \( G \) for Phase I (without side payments), Phase II (with side payments) and the heuristic CACM [3]

<table>
<thead>
<tr>
<th>group</th>
<th></th>
<th>Phase I</th>
<th>Phase I+II</th>
<th>CACM [3]</th>
</tr>
</thead>
<tbody>
<tr>
<td>s2</td>
<td>2</td>
<td>2.1</td>
<td>0.7</td>
<td>2.1</td>
</tr>
<tr>
<td>s5</td>
<td>5</td>
<td>9.1</td>
<td>4.6</td>
<td>9.2</td>
</tr>
<tr>
<td>m2</td>
<td>2</td>
<td>2.2</td>
<td>1.5</td>
<td>2.0</td>
</tr>
<tr>
<td>m5</td>
<td>5</td>
<td>7.8</td>
<td>6.5</td>
<td>8.1</td>
</tr>
</tbody>
</table>

As Table 3 shows, the approach CSP (Phase I+II) achieves on average over all tested groups of instances the best results, i.e., the smallest gap to the best-known central solution. Note, for the group of small instances (s2, s5), the optimal solution for a central decision maker is known.

The collaborative local search approach of Phase I is able to compete with the results of CPMC. For the instance groups s5 and m5 it even achieves slightly superior results. This has to be stressed, because Phase I is quite straightforward compared to the approach
CPMC which embeds collaborative local search in an ant colony metaheuristic framework. The essential point that improves the performance appears to be using the voting based acceptance criterion at frequent intervals.

Phase II integrates side payments into the search. According to Table 3 this clearly pays off. The average gap to the central best-known solution is significantly reduced. For instance group s2, s5, m2, and m5 the gap is reduced by about 66 percent, 50 percent, 32 percent, and 17 percent, respectively. Therefore, integrating side payments directly into collaborative local search approaches is appears to be a promising way to find better, jointly accepted solutions.

5.3. Convergence behavior

Figure 1 shows the convergence behavior for a selected instances (no. 54 from instance group s5) with five agents. A point in Figure 1 represents a solution which is jointly accepted by all five agents. Values of temporarily accepted solutions are not shown. Furthermore, Figure 1 shows the local costs $f_a$ of each agent $a \in A$, side payments $p_a$ or the net effects $f_a - p_a$ are not directly shown. Phase I and Phase II each generate 25,000 solutions.

Because the instance at hand is small, the search of Phase I soon generates solutions which are rejected by at least one agent. That is, the search is locked in a local negotiation optimum. After 25,000 generated solutions at the end of Phase I, the total cost $f(y)$ are 987.48. Phase II starts, now side payments are possible. After a couple of negotiation rounds, the agents agreed on an improved solution with lower local costs for the agents $a_2$, $a_3$, $a_4$, and $a_5$. At the same time, side payments are negotiated, that overcompensate agent $a_1$ for accepting a solution with higher local costs. That is, a fraction of the achieved costs savings has to be paid to agent $a_1$.

During Phase II the total cost $f'$ could be reduced by 25 percent to 739.26 cost units (CU). The local cost of agent $a_1$ increase from 223.48 CU to 395.26 CU which is 171.78 CU or approximately 77 percent. In addition, agent $a_1$ receives a total payment of 214.77 CU by the agents $a_2$ to $a_5$. Therefore, agent $a_1$ is actually overcompensated for accepting inferior local solutions by 45.99 CU. This overcompensation is due to the process of Phase II, in particular the constant cost increment $\Delta_a$ proposed by the agents $a \in A$ which is only known to the mediator. The mediator is neutral and treats the agents as equal, the mediator therefore does not use the information advantage to make counter proposals which might lead to lower side payments and may reduce overcompensation considerably.

6. Conclusion

A way to integrated side payments into a heuristic and collaborative solution approach for the distributed multi-level uncapacitated lot-sizing problem has been presented. Side payments compensate one or more agents for accepting solutions with higher local costs in order to find a solution with lower global costs. The search works on an encoded solution and consists of two phases. Phase I performs simple, but effective collaborative hill-climbing moves. Phase II considers side payments via a modified voting procedure which temporarily accepts non-improving solutions and negotiates side payments that lead to superior solutions for all agents. The approach has been tested on a set of 272 small and medium sized benchmark instances. On these instances, the present approach outperforms on average most existing approaches published previously. However, for a set of larger instances, more powerful neighborhood structures have to be developed. Nevertheless, considering side payments during search will most likely also improve the performance of collaborative search approaches for other problems.
Acknowledgment

The cooperative junior research group on Computational Logistics is funded by the University of Bremen in line with the Excellence Initiative of German federal and state governments.

References


