Stochastic uncapacitated single-item lot-sizing problem: a dual dynamic decomposition approach

Franco Quezada¹, Céline Gicquel², Safia Kedad-Sidhoum³

¹ Sorbonne Université, LIP6, F-75005 Paris, France
franco.quezada@lip6.fr

² Université Paris-Saclay, LRI, F-91190 Gif-sur-Yvette, France
gicquel@lri.fr

³ Conservatoire national des arts et métiers, CEDRIC, F-75003 Paris, France
kedad_sidhoum@cnam.fr

Keywords: Stochastic lot-sizing, Stochastic Dual Dynamic Integer Programming

1 Introduction

The uncapacitated single-item lot-sizing problem (ULS) aims at determining the quantity to be produced in each time period in order to meet demand over a finite discrete-time planning horizon. We devote to investigate an extension of the ULS where demand and costs are subject to uncertainty. We consider a multi-stage decision process corresponding to the case where the value of the uncertain parameters unfolds little by little following a discrete-time stochastic process and the production decisions can be made progressively as more and more information on the demand and cost realizations is collected. In order to address this problem, we rely on a multi-stage stochastic integer programming approach, where the uncertain parameters are represented by a discrete scenario tree.

The complexity of the stochastic uncapacitated single-item lot-sizing problem was studied in [3] and a special case of the problem was proved to be NP-Hard. It is thus unlikely to find algorithms for the problem which are polynomial in the number of time periods $T$.

A dynamic programming algorithm for solving the stochastic ULS was developed in [2]. The algorithm is polynomial in the number of nodes of the scenario tree but this number increases exponentially fast with the number of time periods $T$. On the other hand, several works have been devoted to the polyhedral study of the stochastic ULS on a scenario tree and as a result, several class of valid inequalities have been developed to strengthen the linear relaxation of its extensive mixed-integer linear formulation (see e.g. [1]). These valid inequalities are embedded in a branch-and-cut algorithm and they have proved to be effective solving the stochastic ULS on medium-size scenario tree.

Unfortunately, implicit enumeration methods, such as branch-and-cut algorithms, do not scale up well with the size of the scenario tree. Decomposition methods, such as Stochastic Dual Dynamic Decomposition, are thus an attractive alternative to tackle instances with large-size scenario trees. Recently, [4] proposed a new extension called Stochastic Dual Dynamic integer Programming (SDDiP) of this method in order to solve multi-stage stochastic integer programs with binary state decision variables and non-convex expected cost-to-go functions.

We propose in the present paper to develop a stochastic dual dynamic integer programming approach to solve the stochastic ULS on large scenario trees. We first investigate a stochastic dynamic programming formulation of the stochastic ULS based on continuous state variables. This approach decomposes the original problem into a series of single-node sub-problems which are linked together by dynamic programming equations. As proposed in [4], we reformulate the obtained nodal sub-problems using a binary approximation of the inventory decision variables in order to obtain binary state variables. This allows us to use the SDDiP algorithm to solve
the problem. To the best of our knowledge, this is the first attempt at developing a dynamic programming decomposition approach for the stochastic ULS. We also propose an improved version of the SDDiP algorithm in which a cutting-plane generation phase based on continuous state variables is carried out to build a first approximation of the expected cost-to-go functions. This leads to a two-phase algorithm.

2 Two-Phase SDDiP algorithm

The SDDiP algorithm mainly consist of three steps at each iteration, namely a sampling, a forward and a backward step. First, in the sampling step a subset of scenarios, i.e. a set of paths from the root node to the leaf nodes, are randomly selected. Second, in the forward step a dynamic programming recursion with an approximated expected cost-to-go function is solved for each sampled node. Finally, in the backward step the approximated expected cost-to-go functions are strengthened at each stage $t$ by a set of linear cuts.

In the first phase, we build a first approximation of the expected cost-to-go functions by generating cuts based on a dynamic programming formulation which uses continuous inventory state variables. More precisely, at each node $n$, the nodal sub-problem is reformulated by introducing an auxiliary variable which represents the value of the inventory variable at the parent node of $n$. We then generate strengthened Benders’ cuts to under approximate the expected cost-to-go function as follows:

$$\tilde{\psi}_t^i(\cdot) := \min \{ \theta^t : \theta^t \geq \sum_{m \in C(n)} \rho^{nm}(\tilde{v}_m^l + \tilde{\pi}_m^l s^n) \} \quad \forall l = \{1, ..., i-1\}$$

where $\tilde{v}_m^l$ and $\tilde{\pi}_m^l$ are the coefficients of the cuts generated at iteration $l < i$ and thus $\tilde{\psi}_t^i(\cdot)$ is a piecewise linear under approximation of the expected cost-to-go functions. In these cuts, the value of $\tilde{\pi}_m^l$ is set to the dual value of a copy constraint in the linear relaxation of nodal sub-problem and $\tilde{v}_m^l$ is the optimal value of a suitable Lagrangian relaxation of nodal sub-problem.

This cutting-plane generation strategy relies on the idea that even if the strengthened Benders’ cuts generated by using a relaxation of formulation are not tight, they enable to build a first under approximation of the expected cost-to-go functions with a reduced computational effort as each nodal sub-problem involves a single binary variable $y^n$.

In the second phase, we reformulate the initial dynamic programming formulation by making a binary approximation of the continuous state variables. We then further improve the under approximation of the expected cost-to-go functions by generating three families of cuts, namely integer optimality, Lagrangian and strengthened Benders’ cuts.

Our numerical results show that the initial phase not only significantly improves the quality of the solution found by the algorithm proposed in [4], but also reduces the overall computation time. Furthermore, they show that proposed method outperforms CPLEX 12.8 for large-size scenario tree. Specifically, CPLEX 12.8 reports a relative gap over 44%, whereas our approach reports a relative gap under 13% for large-size instances.

Références


