

# Variants to a knapsack decomposition heuristic for solving R&D pipeline management problems

Brianna Christian, Selen Cremaschi\*

Department of Chemical Engineering, 212 Ross Hall, Auburn University, AL, 36849, United States

## ARTICLE INFO

### Article history:

Received 20 April 2016

Received in revised form 21 October 2016

Accepted 22 October 2016

Available online 24 October 2016

### Keywords:

Multistage stochastic programming  
Pharmaceutical R&D pipeline management  
Decision dependent uncertainty  
Endogenous uncertainty  
Knapsack decomposition algorithm

## ABSTRACT

The knapsack decomposition algorithm (KDA) (Christian and Cremaschi, 2015) decomposes the R&D pipeline management problem into a series of knapsack problems, which are solved along the planning horizon. It yields tight feasible solutions, and improves the solution times by several orders of magnitude for large instances. This paper investigates the impact of problem parameters and size, and KDA decision rules on KDA solution quality and time. The decision rules are (1) timing of new knapsack problem generations, and (2) formulation of the resource constraints in knapsack problems. The results revealed that the KDA decision trees were insensitive to problem parameters, and the KDA solution times grew super-linearly with linear increases in the length of the planning horizon and the number of products. The results suggest that the KDA where knapsack problems are generated after each realization with the original resource constraint yields the most accurate solutions in the quickest time.

© 2016 Elsevier Ltd. All rights reserved.

## 1. Introduction

For new pharmaceutical drug development, the time from discovery to market of a potential molecule has increased considerably over the last decade. This increase leaves limited patent protection for pharmaceutical companies before the generic versions of the drug become available in the market, and places considerable pressure on them to boost the efficiency of their research and development (R&D) pipeline. One way to increase the efficiency is improved management of the R&D pipeline through the scheduling of clinical trials.

The pharmaceutical R&D pipeline planning problem can be characterized by a set of potential new product development projects with a set of resources and expenditures associated with each product. In order for the products to reach the market, a series of clinical trials must be completed. Trials are completed in a stage-gate process where if a product fails to successfully complete a stage, it drops from the portfolio. In this problem, the returns, the necessary resources, and whether or not each product will reach market are not known with certainty when the investments are made. Deci-

sions as to which products to invest in and when to invest in each product are made in order to maximize the returns (Subramanian et al., 2001).

The R&D pipeline management problem is difficult to solve due to the endogenous nature of one of its uncertain parameters, the outcome of the clinical trials. To observe the realizations of this uncertain parameter, the decision to invest in a clinical trial must first occur. This type of uncertainty is called endogenous uncertainty with exogenous realizations, for which the outcome set does not change based on the decisions but the times of realizations depend on the decisions (Mercier and Hentenryck, 2008).

Several methods to solve the pharmaceutical R&D pipeline planning problem have been presented in literature. These methods include both simulation and mathematical programming approaches. In this work, we consider the multistage stochastic programming (MSSP) formulation of the problem developed originally by Colvin and Maravelias (2008). This formulation quickly becomes computationally intractable as the size of the problem increases. Most recent work for solving MSSPs with endogenous uncertainty focuses on developing decomposition strategies and heuristic approaches that reduce the computational expense for handling large-scale problems. In what follows, we provide a brief overview of these approximate solution approaches.

Goel and Grossmann (2004) obtained an approximate solution to the problem by searching a sub-space of the feasible region.

\* Corresponding author.

E-mail addresses: [selen-cremaschi@auburn.edu](mailto:selen-cremaschi@auburn.edu), [selen.aydogan@gmail.com](mailto:selen.aydogan@gmail.com) (S. Cremaschi).

## Nomenclature

### Set

- $i \in I$  The set of items generated by the knapsack algorithm  
 $k \in K_t$  The set of knapsack problems at time  $t$

### Parameters

- $W_{i,r}$  The item weight of item  $i$  for resource  $r$   
 $V_{i,t}$  The item value of item  $i$  at time  $t$   
 $E_{i,t,k}$  Represents the ability to pack item  $i$  at time  $t$  in knapsack problem  $k$   
 $P_r$  The penalty for exceeding resource  $r$   
 $C_r$  The number of resources  $r$  currently invested

### Variables

- $x_i$  The binary decision to pack item  $i$   
 $y_r$  A binary variable indicating whether a penalty is being accessed for resource  $r$   
 $z_{i,r}$  A binary linearization variable

The offshore oil infrastructure planning problem was solved to determine the optimal placement of oil production platforms by minimizing the risk of obtaining a negative net present value. Uncertainty was considered in the production of each well. The sub-space for the approach is defined by removing the scenario dependency of investment decisions for uncertain fields, yielding a more constrained version of the original problem. Then, this constrained version is solved to optimality. The results revealed that the solutions obtained to the constrained formulation were significant improvements compared to the deterministic solutions.

Mercier (2009) proposed a multi-step anticipatory algorithm, which uses a sample average approximation to generate a Markov Decision Process (MDP). The MDP is then solved, and the greedy solution is returned. The algorithm was tested using 12 instances of the pharmaceutical R&D pipeline management problem. The quality of the solution obtained by the algorithm were 10% better than the dynamic programming equivalent for all instances. The authors concluded that the algorithm was, nevertheless, computationally expensive when applied to the pharmaceutical R&D pipeline management problems.

Colvin and Maravelias (2009) explored a rolling-horizon approximation approach, which yielded tight feasible solutions for large instances of the problem. The authors divided the planning horizon into a finite number of subsets. A relaxed MSSP is generated for the first subset by removing all inequality non-anticipativity constraints (NACs) for the stages beyond the first subset. The solution of the relaxed MSSP is implemented for the first subset, and related uncertainty is realized. Then, the process is repeated for each subset until the end of the planning horizon is reached. The authors were able to successfully solve cases with more than 1000 scenarios.

Solak et al. (2010) developed a sample average approximation (SAA) algorithm to solve the R&D project portfolio optimization problem. Smaller MSSPs are generated using a sample of the full scenario set. The upper-bound of these MSSPs are obtained by using their Lagrangean duals and updating the multipliers with a modified sub-gradient optimization algorithm. The lower bound is generated via a heuristic that searches for a feasible solution in the vicinity of the Lagrangean dual solutions obtained during each iteration of the sub-gradient algorithm. The authors suggest that a branch-and-bound algorithm may be used to close the duality gap if necessary. The solutions of the MSSPs are called candidate solutions. The quality of the first-stage decisions of the candidate

solutions are evaluated using a larger sample set of the scenarios. By repeating the process, the variance of the results is used to find a bound on the true solution. Computational studies included two technology portfolio examples where five and 10 projects were considered. The results revealed that the algorithm was able to generate solutions with an estimated optimality gap of around two percent.

Tarhan et al. (2013) presented an approach that improves upper bounds for solving non-convex MINLPs with decision-dependent uncertainty. The algorithm solves the Lagrangean relaxation of the dual problem where NACs have been removed to obtain the upper bound. The lower bound is generated by locating a feasible solution using a rolling-horizon approach. The authors solved two non-convex non-linear problems. The first problem is a version of the process synthesis problem presented in Goel and Grossmann (2006). The second is an offshore oil field planning problem originally presented in Tarhan et al. (2008). The authors concluded that using the outer approximation solution to upper bound the intermediate problems generated during the branch-and-bound algorithm rather than solving them to optimality reduced the solution time of both problems by 60%, and the solutions of the intermediate problems remained within 0.01% of their optimal solutions.

Gupta and Grossmann (2014) improved a Lagrangean decomposition scheme by incorporating a scenario grouping strategy. The scenario grouping strategy allows partial decomposition of the full space model. Groups of scenarios were generated based on differences in the outcomes of uncertain parameters. The grouping strategy was applied to problems previously solved by Goel and Grossmann (2004) and Goel and Grossmann (2006). The authors concluded that their scenario grouping strategy provided a tight bound on the solution in fewer iterations.

Previously, we presented a Knapsack Decomposition Algorithm (KDA) that decomposes a MSSP of pharmaceutical R&D pipeline management problem into a series of knapsack problems (Christian and Cremaschi, 2015). Knapsack problems are generated and solved at the first time period and at time periods where there are no active trials until the end of the planning horizon is reached. The KDA solution yields a feasible decision tree, i.e., an implementable clinical trial schedule, for the original MSSP. We used KDA to solve problems with two, three, five, six, seven, and ten products. The number of scenarios in each of these problems ranged from nine to over one million. These computational studies revealed that the expected net present values (ENPVs) for the decision trees generated by KDA ranged between 0% and 6% of the ENPV of the true solutions, and the KDA yielded these solutions up to 59 times faster than solving the deterministic equivalent of the MSSP formulation using state-of-the-art commercial solvers.

The quality of the KDA solution may depend on the parameters of the R&D pipeline problem, and KDA specific decision rules, i.e., (1) which time period knapsack problems are generated, and (2) how overscheduling in the pipeline is handled. In this paper, we computationally investigate how the solution time and quality of KDA solution change with changes in the size and parameters of the R&D pipeline problem and KDA specific decision rules. The effect of the former is investigated via sensitivity analysis where the problem size and the parameters are perturbed around their original values. For the latter, we present several variations of the KDA obtained by modifying the decision rules, and discuss their impact on the solution times and qualities. Section 2 briefly outlines the original KDA. Details of the sensitivity study is discussed in Section 3. In Section 4, the variations to the KDA decision rules are introduced. Section 5 outlines the computational studies, and Section 6 presents and discusses the results. Section 7 summarizes our findings and discusses future directions.

---

```

1. Generate Items,  $i \in I$ 
2.  $t := 0$ 
3. For  $i \in I$ , Calculate  $W_{i,r}, V_{i,t}, E_{i,t,0}$ 
4.  $\alpha = \text{Knapsack Solution using } (W_{i,r}, V_{i,t}, E_{i,t,0}, W_r^{\max})$ 
5.  $\phi = \text{argmax}(\tau_i + t \ \forall i \text{ s.t. } x_i = 1 \text{ in } \alpha)$ 
6. Generate set  $2^{|\alpha|}$  knapsack problems,  $S$ 
7.  $K_{\phi+t} = K_{\phi+t} \cup S$ 
8.  $t := t + 1$ 
9. While  $t < t^{\max}$ ,
10.    $k := 0$ 
11.   While  $k < |K_t|$ ,
12.     For  $i \in I$ , Calculate  $W_i, V_{i,t}, E_{i,t,k}$ 
13.      $\alpha = \text{Knapsack Solution using } (W_{i,r}, V_{i,t}, E_{i,t,k}, W_r^{\max})$ 
14.      $\phi = \text{argmax}(\tau_i + t \ \forall i \text{ s.t. } x_i = 1 \text{ in } \alpha)$ 
15.     Generate set  $2^{|\alpha|}$  knapsack problems,  $S$ 
16.      $K_{\phi+t} = K_{\phi+t} \cup S$ 
17.      $k := k + 1$ 
18.    $t := t + 1$ 

```

---

Fig. 1. Pseudocode for the KDA algorithm.

## 2. The knapsack decomposition algorithm (KDA)

The KDA was developed to solve the pharmaceutical R&D pipeline management problem. The pharmaceutical R&D pipeline management problem is characterized by the following,

Sets	Description
$d$ -CD	A set of new drug development projects
$j$ -CJ	A set of clinical trial
$r$ -CR	A set of resources
$t$ -CT	A set of discrete time periods
Parameters	
$\gamma^L$	Coefficient for loss of active patent life
$\gamma^D$	Coefficient for loss of products idle in the pipeline
$Rev_d^{\max}$	The maximum revenue of product $d$ for completing all trials
$\rho_{d,j,r}$	The resource $r$ requirements for trial $j$ of product $d$
$c_{d,j}$	The monetary cost of starting trial $j$ for product $d$
$\tau_{d,j}$	The number of time periods required to complete trial $j$ for product $d$
$\rho_r^{\max}$	The maximum number of resource $r$ available

Each new drug development project is required to complete each of the clinical trials. If a drug-development project successfully completes the entire set of clinical trials the revenue associated with that drug is realized. Decisions are made as to which trial(s) to start for which drug(s) at which time period(s). There are limited resources available for investment at each time period. The objective is to maximize the ENPV of the clinical trial plan. Penalties are incurred for drugs remaining idle in the pipeline and for loss of active patent life. The MSSP model of this planning problem can be found in [Colvin and Maravelias \(2008\)](#).

The KDA was developed as a heuristic approach for finding feasible solutions for MSSPs using a series of knapsack problems. Knapsack problems are generated along the planning horizon to form the heuristic solution. Pseudocode for the algorithm can be seen in [Fig. 1](#). The first step in the KDA is to generate a set of potential items,  $i \in I$ . Items in the KDA are generated based on the decision variable(s). For the pharmaceutical R&D pipeline management problem, the decision variable is when to start each clinical trial. Here we represent each product clinical-trial pair as an item.

At time zero, the weight and value of each item is calculated (Step 3). These values are calculated using Eqs. (1) and (2). The weight of each item,  $W_{i,r}$ , is simply the resources required to start the corresponding product trial. Each item will have  $|R|$  weights, one for each resource.

$$W_{i,r} = \rho_{d(i),j(i),r} \ \forall r \in R \quad (1)$$

The value of an item,  $V_{i,t}$ , is calculated based on the expected gains of starting a clinical trial. In this case, we have used the expected revenue of the product deducting penalties for loss of active patent

life. The revenue is weighted by the probability that the product will complete the current and remaining clinical trials.

$$V_{i,t} = \left[ Rev_{d(i)}^{\max} - \gamma_{d(i)}^L \left( t + \sum_{j' \geq j(i)} \tau_{d(i),j'} + 1 \right) \right] \prod_{j' \geq j(i)} p_{d(i),j'} \quad (2)$$

The value for  $E_{i,t,0}$  is calculated using a series of logical tests (Step 3).  $E_{i,t,0}$  takes a value of one if an item has not been packed in a previous knapsack and if the item corresponds to the first trial of a product (i.e.  $j(i) = 1$ ). Similarly,  $E_{i,t,0}$  will take a value of one if the items corresponding to the pre-requisite trials have been packed and the uncertainty in each of the pre-requisite trials has been revealed. At  $t = 0$ , it is often the case that the only items corresponding to the first trial of each product have values of  $E_{i,t,0}$  equal to one.

The algorithm continues by generating an initial knapsack problem (Step 4). The formulation for the knapsack can be seen in Eq. (3).

$$\max \sum_{i \in I} V_{i,t} x_i \quad (3)$$

$$\sum_{i \in I} W_{i,r} x_i \leq W_r^{\max} \ \forall r \in R$$

$$\sum_i \left[ \sum_{j' \geq j(i)} \rho_{d(i),j',r} \tau_{d(i),j'} \right] x_i \leq \max \left\{ \sum_{j' > j(i)} \tau_{d(i),j'} + 1 \ \forall i \text{ s.t. } E_{i,t,k} = 1 \right\} \rho_r^{\max} \ \forall r \in R$$

$$x_i \leq E_{i,t,k}$$

$$x_i \in \{0, 1\}$$

Much like the general knapsack formulation the objective of the knapsack problem in the KDA is to maximize the value of items packed ( $x_i = 1$ ) in the knapsack constrained by a maximum weight limit. Unlike the general knapsack problem, the KDA knapsack problem introduces a resource-overscheduling constraint. The resource over-scheduling constraint prevents items from being packed in the knapsack if there are not enough resources available to pack the items corresponding to subsequent trials continuously. This constraint is aimed at reducing the number of products left idle in the development pipeline. The solution of the knapsack problem is used to determine the value of the decision variables at the time the knapsack was solved. For the pharmaceutical R&D problem at  $t = 0$ , items selected by the knapsack ( $x_i = 1$ ) correspond to the decision variable at  $t = 0$  which take the value of one.

Steps 5 and 6 determine when and how many new knapsack problems to generate. New knapsack problems are generated after all uncertainty associated with items selected in the initial knapsack problem has been realized. In the case of the pharmaceutical R&D pipeline management problem, new knapsacks are generated after the outcomes of all started clinical trials are realized. The number of new knapsack problems generated is based on the number of items packed in the initial knapsack problem. A new knapsack problem is generated for each possible realization of uncertainty. For instance, if two items were selected in the knapsack and each item had three possible outcomes,  $2^3$  knapsack problems would be generated. In the pharmaceutical R&D pipeline management problem, the outcomes of uncertainty are either pass or fail. Therefore, for each item packed in the knapsack there are two possible realizations. The total number of new knapsack problems generated is equal to  $2^{|\alpha|}$ , where  $\alpha$  is the number of items packed in the knapsack. Step 7 adds the newly generated knapsack problems to the set of knapsack problems at time  $t + \phi$ ,  $K_{t+\phi}$ . The time is incremented by one ( $t := t + 1$ , Step 8), and knapsack problems in  $K_t$  are solved.

For each problem in  $\mathbf{K}_t$  new values for  $W_{i,r}$ ,  $V_{i,t}$ , and  $E_{i,t,k}$  are calculated (Step 12).  $E_{i,t,k}$  is calculated using the knowledge of realized values of uncertainty and previously selected items to employ the same logic applied to  $E_{i,t,0}$ . Values calculated in Step 12 are used with the knapsack formulation shown in Eq. (3). The items selected in the knapsack represent the decision variables at time  $t$  which take the value of one in scenarios where the realizations match the knapsack problem  $k$ . For each knapsack problem solved in  $\mathbf{K}_t$ ,  $2^{|\alpha|}$  new knapsacks are generated at time  $\phi + t$  (Steps 14–16). After all knapsack problems in  $\mathbf{K}_t$  are solved, the time is incremented again. The process of solving and generating new knapsack problems continues until the end of the planning horizon is reached. The detailed description of the KDA is presented in Christian and Cremaschi (2015).

### 3. Evaluating sensitivity of the KDA solution quality and time to original problem parameter values and size

The parameters of the pharmaceutical R&D pipeline management problem are the lengths and the costs of the clinical trials, the revenue realized after successful completion of all clinical trials, and the penalty factors associated with delay of products already in the pipeline and loss of patent life. We test the sensitivity of the KDA solution to the values of these parameters by perturbing each one individually. The size of the R&D pipeline management problems changes with the number of trials, the length of the planning horizon, and the number of resources. To study the impact of problem size on the KDA performance, we constructed a set of problems where the number of products, the number of trials, the length of the planning horizon, and the number of resources are varied independently.

### 4. Variants of KDA decision rules

The KDA has two decision rules. The first one determines when new knapsack problems are generated. Our previous computational studies revealed that although the KDA yielded tight feasible solutions (within three percent), for some problems, particularly the ones with large differences in clinical trial lengths, it generated very sparse decision trees. For these problems, the quality of the solution was worse (closer to three percent). We hypothesize that the sparse decision tree may be a result of the algorithm requiring that all started trials must be completed prior to starting new trials.

The second decision rule specifies the constraint that aims to prevent products from being idle in the pipeline. We refer to this constraint as the resource overscheduling constraint. The original KDA formulation introduces a hard resource overscheduling constraint, which ensures that for every item packed there will be sufficient number of resources to continue subsequent trials. Because this constraint does not consider the possibility that a product may fail a trial, it may significantly limit future investments in additional products, and hence, may lead to sparse decisions trees.

#### 4.1. Sub-problem generation rules

We propose two additional approaches for determining when knapsack problems are generated: (1) at Each Time Period (ETP), and (2) After Each Realization (AER). The ETP generates knapsack problems at each time period where there are idle resources and clinical trials that can be started. When a realization occurs, i.e., one or more of the started clinical trials are completed, new knapsack problems are generated for each realized value. If any items remain in the knapsack (trials that have been started but not yet completed), they are passed as already selected items to each newly generated knapsack problem. Fig. 2 graphically depicts the knapsack generation schemes using ETP (Fig. 2(a)), the original KDA

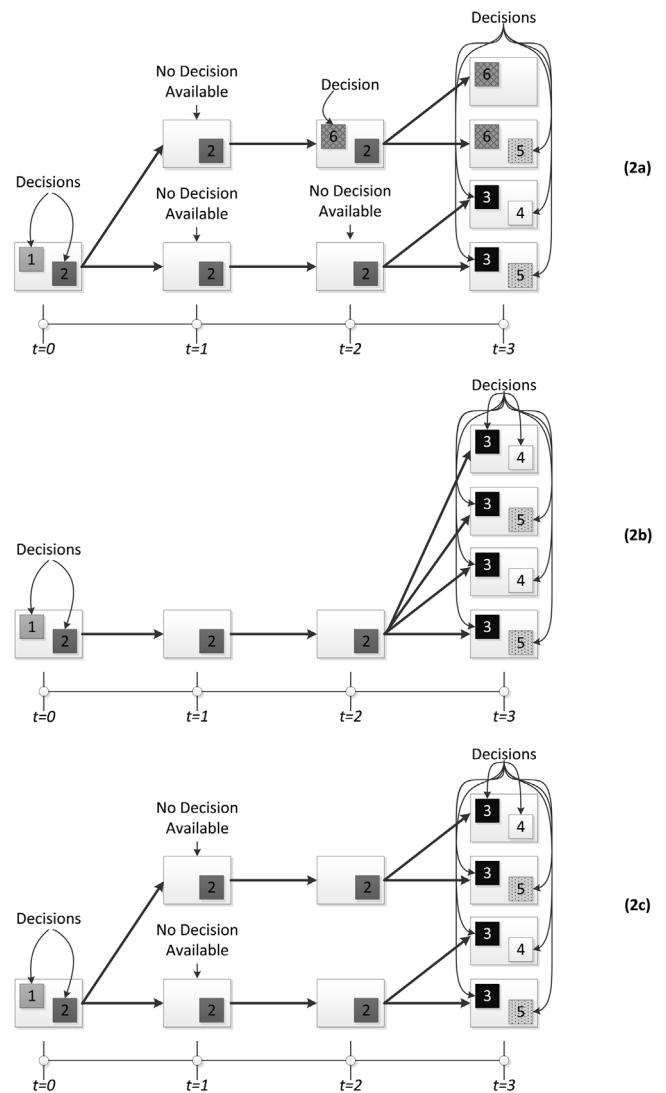


Fig. 2. The knapsack generation schemes, (a) ETP, (b) Original KDA, (c) AER.

(Fig. 2(b)), and AER (Fig. 2(c)) for three time periods. The solution of the first knapsack problem is to pack items 1 and 2 (Fig. 2,  $t = 0$ ). The trial associated with item 1 is completed at  $t = 1$ , while the trial of item 2 is completed at  $t = 3$ . The ETP approach generates two new knapsack problems at  $t = 1$  (Fig. 2(a)). Each knapsack problem corresponds to a unique realization associated with item 1. The binary associated with item 2 is set equal to one ( $x_2 = 1$ ) in both knapsack problems. For this example, we assume that there are not sufficient resources to add any more items at  $t = 1$ . At  $t = 2$ , there are no realizations, i.e., the trial associated with item 2 is not completed. The ETP algorithm generates and solves two new knapsack problems. In each knapsack problem, the value of  $x_2$  is set equal to one. The solutions of these knapsack problems are different. In one case, the algorithm selects item 6. In the other case, the algorithm does not have sufficient resources to add another item. The uncertainty realization associated with item 2 occurs at  $t = 3$ . As can be seen in Fig. 2(a), the KDA generates two knapsack problems for each branch. In the case where item 6 is selected, the knapsack problem corresponding to one realization is able to add item 5. The other knapsack problem does not have sufficient resources to add another item.

Fig. 2(b) depicts the knapsack problem generation rule used in the original KDA. The solution at  $t = 0$  selects items 1 and 2. The original KDA does not generate any knapsack problems until the



clinical trials associated with both items are completed, i.e., until  $t = 3$  (Fig. 2(b)). At  $t = 3$ , the original KDA generates and solves four new knapsack problems, each one corresponding to one of the possible outcomes of uncertainty.

The AER approach (Fig. 2(c)) generates and solves knapsack problems when any of the selected trials is completed, i.e., the outcome of an uncertain parameter is realized. Unlike the original KDA, if more than one item were packed in the knapsack, the AER approach generates new knapsack problems at the completion of the shortest clinical trial. Similar to ETP approach, if there are any remaining items in the knapsack (trials that have been started but not yet completed), they are passed as already selected items to each newly generated knapsack problem.

Fig. 2(c) shows that the solution of the first knapsack problem is the selection of items 1 and 2 at  $t = 0$ . The uncertainty associated with item 1 is realized at  $t = 1$ . The AER approach generates new knapsack problems for each realization. Similar to the ETP approach, the AER approach finds that there are insufficient resources to start any new trials. Unlike the ETP approach, the AER approach does not generate new knapsack problems unless there is a realization. Therefore, no knapsack problems is generated at  $t = 2$ . At  $t = 3$ , four knapsack problems are generated, two for each branch. The two knapsack problems in each branch represent the possible realizations of uncertainty associated with item 2. Solutions for each of the four knapsack problems at  $t = 3$  are shown in Fig. 2(c).

#### 4.2. Formulations for resource overscheduling constraint

The original resource overscheduling constraint in the KDA does not allow for items (drug-trial pairs) to be packed as part of the solution if there will not be enough resources to pack items corresponding to subsequent trials of the same drug. For instance, assume that two potential drugs need to complete two trials before reaching market, the duration of all trials are equal, and the resource costs are 10 and 30 for drug A trial one and trial two, 20 for both trials for drug B, and the maximum amount of available resources is 30. The first trials of products A and B cannot be started at the same time because the number of resources needed to start the second trials exceeds the maximum amount. This constraint leads to a conservative solution because it does not consider the possibility of a drug failing to successfully complete the selected trial or the subsequent trials when trying to anticipate the future resource requirements.

We present two new formulations for avoiding possible resource overscheduling. The first one modifies the knapsack problem formulation by adding a penalty term to the objective function rather than an additional constraint. The penalty term grows proportional to the number of resources that exceed the number of available resources, and is shown in Eq. (4).

$$P_r = \lambda \sum_i \left[ \sum_{j \geq j(i)} \rho_{d(i),j,r} \tau_{d(i),j} \right] x_i - \left( \max \left\{ \sum_{j' > j(i)} \tau_{d(i),j'} + 1 \forall i \in \mathbf{E} \right\} \rho_r^{\max} - C_r \right) \quad (4)$$

The amount each resource in excess affects the penalty,  $P_r$ , is given by a rate constant  $\lambda$ . Penalties are only incurred if the number of resources used exceeds the number of resources available. The number of resources in excess of the available resources is calculated by subtracting the maximum resources available  $\left( \max \left\{ \sum_{j' > j(i)} \tau_{d(i),j'} + 1 \forall i \in \mathbf{E} \right\} \rho_r^{\max} \right)$  from the resources allo-

but have not been completed,  $C_r$ , and resources allocated for newly packed items. To enforce that the penalty is only imposed when the number of resources is exceeded, we introduce a disjunction with a binary variable,  $y_r$ , which is one when the number of resources used exceeds the number of resources available and zero, otherwise. The disjunction can be seen in Eq. (5).

$$\left[ \begin{array}{c} y_r \\ \text{Penalty} = P_r \end{array} \right] \vee \left[ \begin{array}{c} -y_r \\ \text{Penalty} = 0 \end{array} \right] \quad (5)$$

We use big-M formulation to convert this disjunction. The objective function with the penalty term is given in Eq. (6).

$$\max \left\{ \sum_i V_i x_i - \sum_r P_r y_r \right\} \quad (6)$$

Expanding the penalty term in Eq. (6), we can see that first term in Eq. (7) is non-linear due to the multiplication of the binary variables  $y_r$  and  $x_i$ .

$$\sum_r P_r y_r = \sum_r \lambda \sum_i \left[ \sum_{j \geq j(i)} \rho_{d(i),j,r} \tau_{d(i),j} \right] x_i y_r - \left( \max \left\{ \sum_{j' > j(i)} \tau_{d(i),j'} + 1 \forall i \in \mathbf{E} \right\} \rho_r^{\max} + C_r \right) y_r \quad (7)$$

We linearize this term by introducing a new binary variable  $z$  where  $z_{i,r} = I_{t_i} \cdot y_r$  and adding the following constraints, Eq. (8), to the knapsack problems.

$$\begin{aligned} z_{i,r} &\leq x_i \forall i, r \\ z_{i,r} &\leq y_r \forall i, r \\ z_{i,r} &\geq y_r + x_i - 1 \forall i, r \end{aligned} \quad (8)$$

The second formulation replaces the original resource overscheduling constraint with a probabilistic constraint, which uses the probability that resources will be needed. The probability that a drug will require resources for a given trial is equivalent to the probability that the drug passes the previous trial(s). The probability that a drug passes a set of trials can be calculated using the probabilities of success in each individual trial. The probability of a series of events occurring,  $Pr$ , where the outcome of each event ( $e_1, e_2, \dots, e_N$ ) is assumed to be independent, can be written as  $Pr = P(e_1 \cup e_2 \cup \dots \cup e_N) = \prod_{i=1}^n P(e_N)$ . Assuming the trial outcomes are independent, the probability that a drug will pass a set of trials is calculated as the geometric sum of the probability of each trial's success. Eq. (9) shows the modified constraint.

$$\begin{aligned} &\sum_i \left[ \rho_{d(i),j(i),r} \cdot \tau_{d(i),j(i)} + \sum_{j' > j(i)} \left[ \prod_{j' > j' > j(i)} P(j' = \text{Pass}) \right] \rho_{d(i),j',r} \cdot \tau_{d(i),j'} \right] \\ &\leq \max \left\{ \sum_{j' > j(i)} \tau_{d(i),j'} + 1 \forall i \in \mathbf{E} \right\} \rho_r^{\max} \forall r \in \mathbf{R} \end{aligned} \quad (9)$$

Notice that the new constraint is a relaxation of the original constraint. Resources for the current trial are allocated ( $\rho_{d(i),j(i),r} \cdot \tau_{d(i),j(i)}$ ) but the resources for subsequent trials are weighted using

the probability that all of the previous trials are successful, i.e.,

$$\left( \sum_{j>j(i)} \left[ \prod_{j>j'(i)} P(j' = \text{Pass}) \right] \rho_{d(i),j,r} \cdot \tau_{d(i),j} \right).$$

## 5. Computational studies

The first set of computational studies investigates the impact of the problem parameters on the KDA performance. The parameters are the trial cost, the revenue for successful completion of the pipeline, the penalties for loss of patent life and non-investment in products currently in the pipeline, the length of each clinical trial, and the overall resource availability. We use the original two-product (2.2.5.2), three-product (3.3.12.2), four-product (4.3.6.2), and five-product (5.3.6.2) cases as base case problems. File names in this work provide descriptive information about the problem solved. For instance, the file name 4.3.6.2 represent the four product base case problem. This problem has four products, three clinical trials, a six time-period planning horizon, and two limiting resources. The base case problems we consider in this work range in number of products from two to five. In all but the two-product case, three trials are required to be completed. The two-product case requires the completion of only two clinical trial. Each case has two resources constraining investment decisions. Planning horizons in each base case range from five time periods and 12 time periods. The parameters for each of the base case problems is given in [Appendix A](#).

The values of the cost, revenue, and penalty parameters are perturbed by  $\pm 10\%$  and  $\pm 25\%$  for each case. The sensitivity of the KDA performance to the lengths of the clinical trials is studied by extending the length of each trial one and two time periods. To study the impact of overall resource availability, we construct four problems with varying degrees of overall resource constraints: (1) unconstrained, (2) 40 percent unconstrained, (3), 70 percent unconstrained, and (4) fully constrained. The base case problems are assumed to be fully resource constrained. The unconstrained case provides enough resources for each product to be completed simultaneously without delay. The number of resources in the 40 percent unconstrained case is calculated by increasing the available resources of the fully constrained case by 40 percent of the difference between the available resources in the unconstrained and the fully constrained cases. In the 70 percent unconstrained case, this increase is 70%.

The second set of computational studies investigates the effect of the problem size on the KDA solution times. For each of our base case problems, we increase the size of the problem by increasing the number of trials, the length of the planning horizon, and the number of resources. [Table 2](#) summarizes the considered variations. A plus sign (+) in [Table 1](#) refers to an increase in magnitude, and a minus sign (–) refers to a decrease. The number next to the sign indicates the magnitude of the increase/decrease. A total of 124 problems were developed to test the sensitivity of the KDA to parameter values and problem size. Information for a specific problem is available from the authors by request.

We tested the performance of knapsack generation approaches (ETP and AER) using the six base case problems. The resource overscheduling constraints were analyzed using the original six problems.

Performance of the KDA is evaluated based on its solution quality and computational time requirements. The quality of the solution is assessed by comparing the KDA solution objective function value with that of the deterministic equivalent of the rigorous MSSP solution. They are expressed as the percent difference from the rigorous MSSP solution. The computation times are given in CPU seconds. Because the KDA generates solutions orders of magnitude

**Table 1**

Problem specifications used for studying the sensitivity of KDA solution time to problem size.

Variation	Variation Value
Number of Trials	+1
	+2
	+3
Number of Resources	+1
	+2
	+3
Length of Planning Horizon	–1
	+1
	+2
Trial Cost	+3
	–25%
	–10%
Active Patent Life Loss Penalty	+10%
	+25%
	–25%
Idle Product Penalty	–10%
	+10%
	+25%
Trial Duration	–25%
	–10%
	+10%
Percent Unconstrained	+25%
	+1
	+2
	0%
	40%
	70%

faster compared to the time it takes to obtain the solution for the rigorous MSSP using commercially available solvers, the computation times for the variants of the KDA are compared to that of the original KDA.

The KDA and the deterministic equivalent formulation of the rigorous MSSP have been implemented using python 3.5 with Pyomo 4.1 (Sandia Corporation, 2013) on Auburn Hopper Cluster. Pyomo solves each knapsack sub-problem using CPLEX 12.63 to an optimality gap of 0.1%. The rigorous MSSPs are solved to an optimality gap of 1% for the two-, three-, and four-product variations and a gap of 5% for the five- and six-product variations. In both cases, the rigorous MSSP is solved using CPLEX 12.63.

## 6. Results and discussion of computational studies

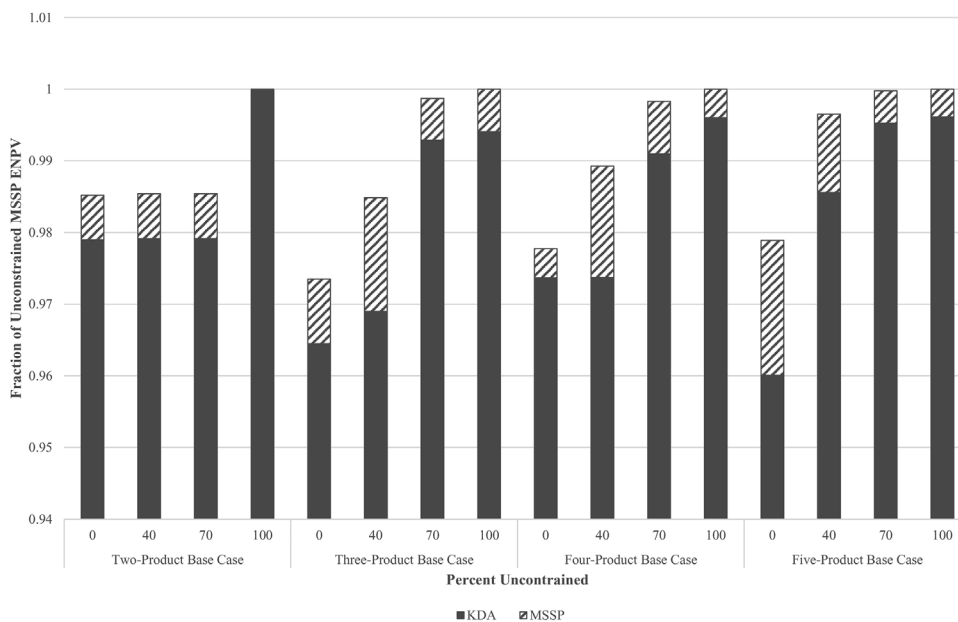
### 6.1. The impact of changes in the R&D pipeline management problem parameters

The computational experiments revealed that the changes in the trial cost(s), the revenue(s), and both penalty parameters had very little effect on the decision trees generated by the KDA. Changes in the parameters, however, (as expected) resulted in fairly large changes in the value of the objective function. The numerical results of these studies are compiled in [Appendix B](#). Similarly, variations in length of each clinical trials did not have a significant impact on the solution quality.

[Fig. 3](#) plots the fraction of the ENPV of the KDA and MSSP for each of the trials vs the ENPV of the fully unconstrained case for each of the base cases. One can observe that the rigorous MSSP ENPVs asymptotically approach the unconstrained solution. A similar behavior is observed for the KDA solutions. However, the KDA solution asymptotically converges to a solution that has a lower ENPV than the MSSP in all but the two-product base case ([Fig. 3](#)). This behavior is an artifact of the knapsack problem generation schema. In the original KDA, knapsack problems are only generated after all realizations occur. Therefore, unless all clinical trials have the same duration penalties are accessed for products sitting

**Table 2**  
Solution times and percent error for original knapsack, AER, and ETP knapsack generation schemes.

	MSSP		Original KDA		Every Time Period (ETP)		After Every Realization (AER)	
	ENPV	Solve Time (CPU sec)	Percent Gap	Solve Time (CPU sec)	Percent Gap	Solve Time (CPU sec)	Percent Gap	Solve Time (CPU sec)
two-product	1110	0.02	1.17	0.05	1.17	0.09	1.17	0.07
three-product	1189	0.84	0.93	0.37	5.05	0.96	1.51	0.35
four-product	1683	2.71	0.42	0.81	0.30	1.26	0.30	0.87
five-product	2083	16.95	1.92	1.24	1.44	2.12	1.44	1.7
six-product	2412	97.67	0.37	1.63	0.21	3.18	0.21	2.45
seven-product	–	–	–	8.17	–	21.33	–	16.67
Average			0.96		1.63		0.93	



**Fig. 3.** The change in ENPV of the KDA and MSSP solutions with resource availability.

idle in the pipeline, and the resulting ENPV is lower than the ENPV of the rigorous MSSP solution.

In our case studies, we also investigated how the size of the problem impacts the quality of the KDA solution. When the length of the planning horizon and the number of resources were varied the KDA remained within five percent of the rigorous MSSP solution. However, when the number of trials was increased the KDA produced solutions greater than 30% lower. These results can be seen in Appendix B. After examining the decision trees for both the KDA and the rigorous MSSP solutions where the difference in ENPVs exceeded five percent, we discovered that the solution for the MSSP was to “do nothing”. Because the knapsack problems generated by the KDA only includes positive value items (see Eq. (2)), the KDA cannot yield the “do nothing” solution. Therefore, the quality of the KDA solutions will deteriorate quickly if the optimum decision set is empty.

## 6.2. The impact of R&D pipeline management problem size on KDA solution time

Fig. 4 plots how KDA solution times and the number of knapsack problems solved change with number of resources, number of clinical trials, and length of planning horizon. The charts on top row of Fig. 4 plot the KDA solution times versus the variation in the number of resources, number of clinical trials, or length of planning horizon. These charts reveal that the number of resources and clinical trials have negligible impacts on KDA solution times. The

number of resources affects the number of weight constraints in the knapsack problems. Adding weight constraints increases the complexity of an individual knapsack problem, and hence, may increase its solution time. For the level of variation considered in the number of resources for this computational study, the solution times for individual knapsack problems did not change significantly. The number of trials affects the total number of items in knapsack problems, but does not change the number of items that can be packed in any given time. The maximum number of items that may be packed in any knapsack is equal to the number of products.

The KDA solution times appear to grow significantly when the length of the planning horizon is increased. (Fig. 4). The increase in planning horizon corresponds to an increase in potential decision points. An increase in decision points increases the number of knapsack problems solved, and hence, the KDA solution time.

The charts on the bottom row of Fig. 4 plot the number of knapsack problems generated versus the variation in the number of resources, number of trials, and length of the planning horizon. One of the trends revealed in these plots is the positive correlation between the KDA solution time and the number of knapsack problems solved. The trend is most noticeable when comparing the solution time and the number of knapsack problems solved when the length of the planning horizon is varied. The knapsack problem is an np-complete problem, and, hence, the solution time for each knapsack problem increases in non-polynomial time based on the number of items and the dimensionality of the problem (number of weight constraints). The KDA solution time is equal to the cumula-

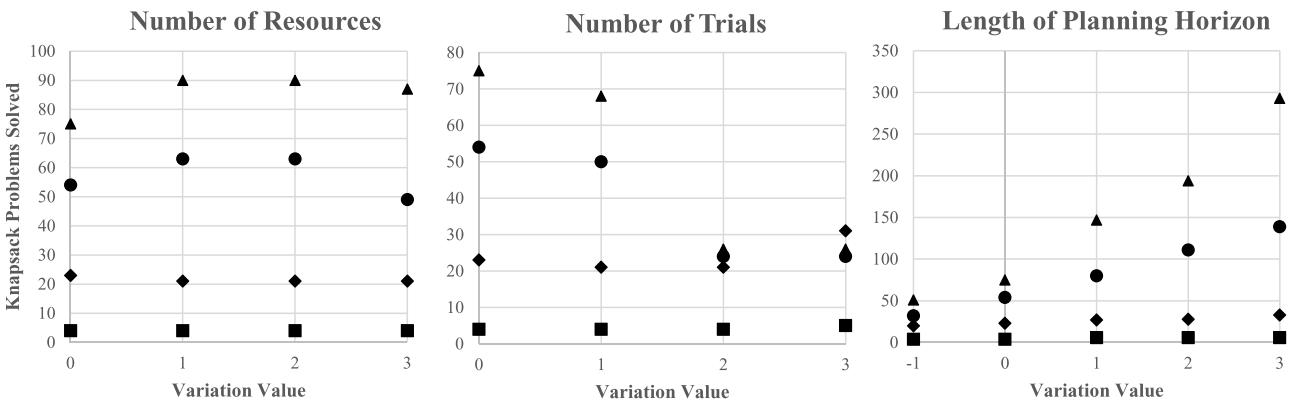
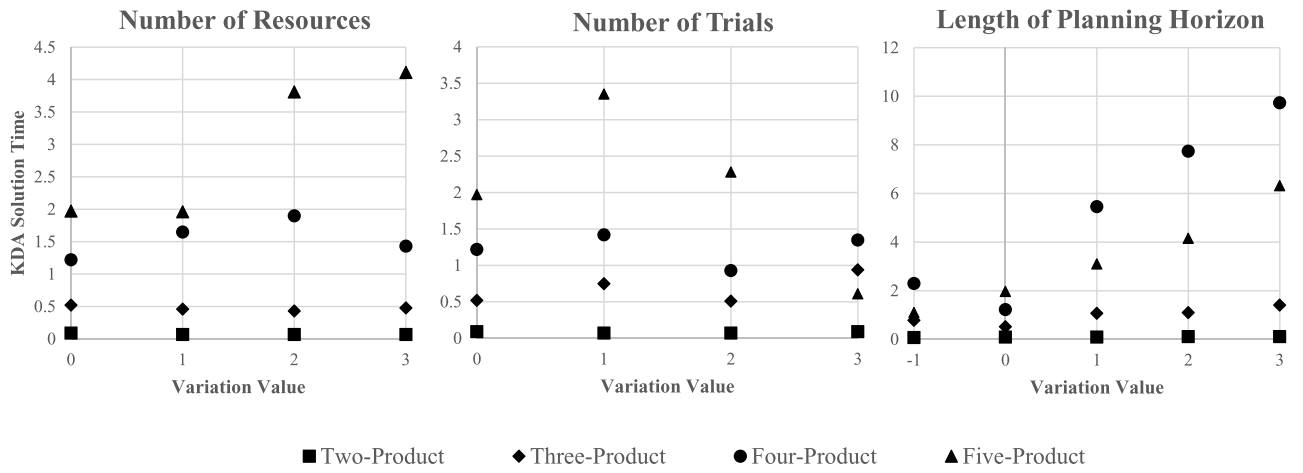


Fig. 4. The impact of problem size on the KDA solution time and the number of knapsack problems.

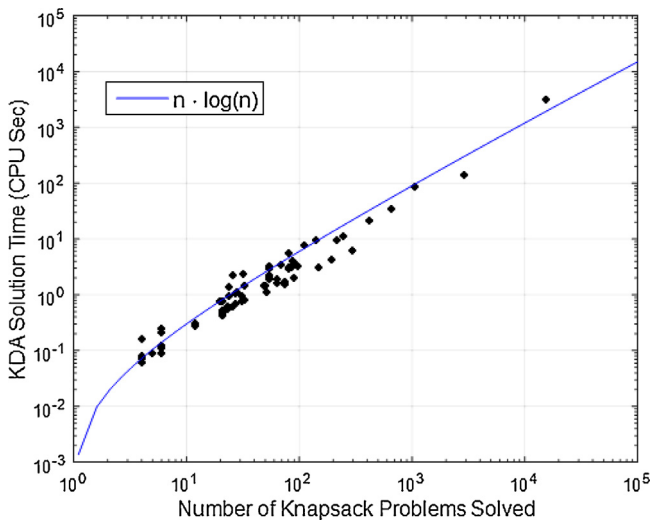


Fig. 5. The correlation between number of knapsack problems solved and the solution time in CPU seconds of the KDA.

tive solution time of all knapsack problems plus the additional time consumed for logic operations (e.g., for determining eligible items). Fig. 5 plots the KDA solution time against the number of knapsack problems solved. The plot suggests that the KDA time complexity is linearithmic ( $O(n \log(n))$ ) where  $n$  is the number of knapsack problems solved. In Christian and Cremaschi (2015), we presented a loose theoretical upper-bound on the number of knapsack problems that may be solved. For all problems considered in

this computational study, the actual number of knapsack problems solved are significantly lower than the theoretical bound. Improving the theoretical bound on the number of knapsack problems solved will allow for better prediction of KDA solution times.

The number of knapsack problems generated depends on the number of realizations that occur in the planning horizon and the number of items. The number of realizations in the planning horizon is affected by the length of the planning horizon, and the duration of each clinical trial that is selected. A longer planning horizon translates to more realizations and thus more knapsack problems. Quantifying the number of realizations also requires the knowledge of packed items which is not known *a priori*. Each item packed in the knapsack has a corresponding trial duration. The duration for each trial is not guaranteed to be identical, therefore more realizations will occur when items with short trial durations are packed. The packed items in knapsack problems are limited due to resource constraints. However, resource constraints only constrain the number of resources available not the number of items packed thus some knapsacks may have many light (low resource) items while others just a few heavy (high resource) items. The bound presented in Christian and Cremaschi (2015) takes a very conservative approach to addressing each of these complications. The bounding approach assumes each knapsack packs a maximum number of items based on the items weights (resource requirements) and the maximum number of resources available. It also assumes that the same knapsack is packed at each realization regardless of the availability of items to pack and outcomes of items. Realizations  $\zeta$  are assumed to occur every  $\min(\tau_{d,j} \forall d, j)$  time periods. These assumptions cause the bounding approach to calculate the number of knapsacks solved if you could pack the



---

```

A = ∅
tp = 0
EKS = 1
while tp < |T| do
    C = I \ A
    B = solution(max(∑xi s. t. ∑wixi ≤ ρmax ∀i ∈ C))
    A = A ∪ B
    tp = tp + min(τd(i),j(i) ∀i ∈ B)
    EKS = EKS + EKS · 2|B|
end

```

---

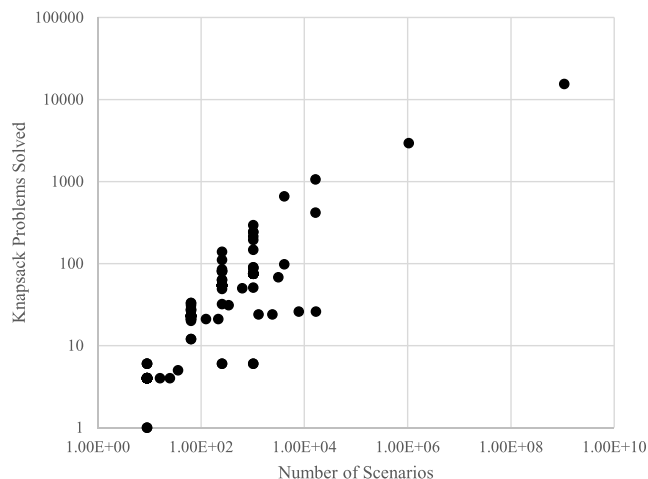
**Fig. 6.** An algorithm to estimate the number of knapsack problems solved by the KDA.

maximum number of items allowed by the available resources in every knapsack at each realization.

Here, we introduce an algorithm that improves the bound by considering that each branch in the decision tree is limited to packing each item once. It assumes that realizations occur when the first item in the knapsack is completed. Similar to the previous approach, we do not consider the value of the uncertain parameter after it has been realized. We assume that subsequent items can be packed whether or not the clinical trial is successful. This assumption creates an upper bound on the number of knapsack problems solved. The algorithm is presented in Fig. 6.

The algorithm calculates the estimated number of knapsacks (EKS) by stepping through the planning horizon. At each step, the algorithm finds the maximum number of items that can be packed assuming that items that have been packed previously are no longer eligible. Using the algorithm, we predicted the number of knapsack problems for the original set of problems presented in Christian and Cremaschi (2015). For the two-product case, the algorithm predicted a total of five knapsack problems. Using the KDA approach, the number of knapsacks solved was four. The difference is due to the generation of knapsack problems regardless of the trial outcomes. In the two-product case, the number of failures that result in counting fictitious knapsack problems is low. This count increases with the size of the problem (number of products and trials). Therefore, as the size of the problem increases the estimate provided by Fig. 6. degrades. For the three product case, the ratio of the number of actual knapsack problems solved to the estimated number is 0.20. For the largest problem (the ten-product case), the ratio is the lowest at 0.05.

Another measure of problem size is the number of scenarios, and Fig. 7 shows the change in the number of knapsack problems solved with the number of scenarios. Notice that the general trend of Fig. 7 shows an increase in the number of knapsack problems solved as the number of scenarios increases. The number of scenarios does not directly impact the number of knapsack problems solved, however the number of scenarios in a problem does depend on the number of uncertain parameters and the number of realizations for each uncertain parameter. The number of uncertain parameters in pharmaceutical R&D pipeline management problem depends on the number of products. For each additional product, there is one additional uncertain parameter, and for each additional clinical trial, there is one additional realization per uncertain parameter. For each additional uncertain parameter, there are  $(|J| + 1)$  times more scenarios. Similarly, for each realization, there are  $\left(\frac{|J|+1}{|J|}\right)^{|D|}$  times more scenarios. Hence, both the number of knapsack problems and the number of scenarios increases with increases in number of products, and hence, the positive trend in Fig. 7. Unlike increasing the products, increasing the number of clinical trials



**Fig. 7.** The number of knapsack problems solved using the KDA plotted against the number of scenarios in the equivalent MSSP.

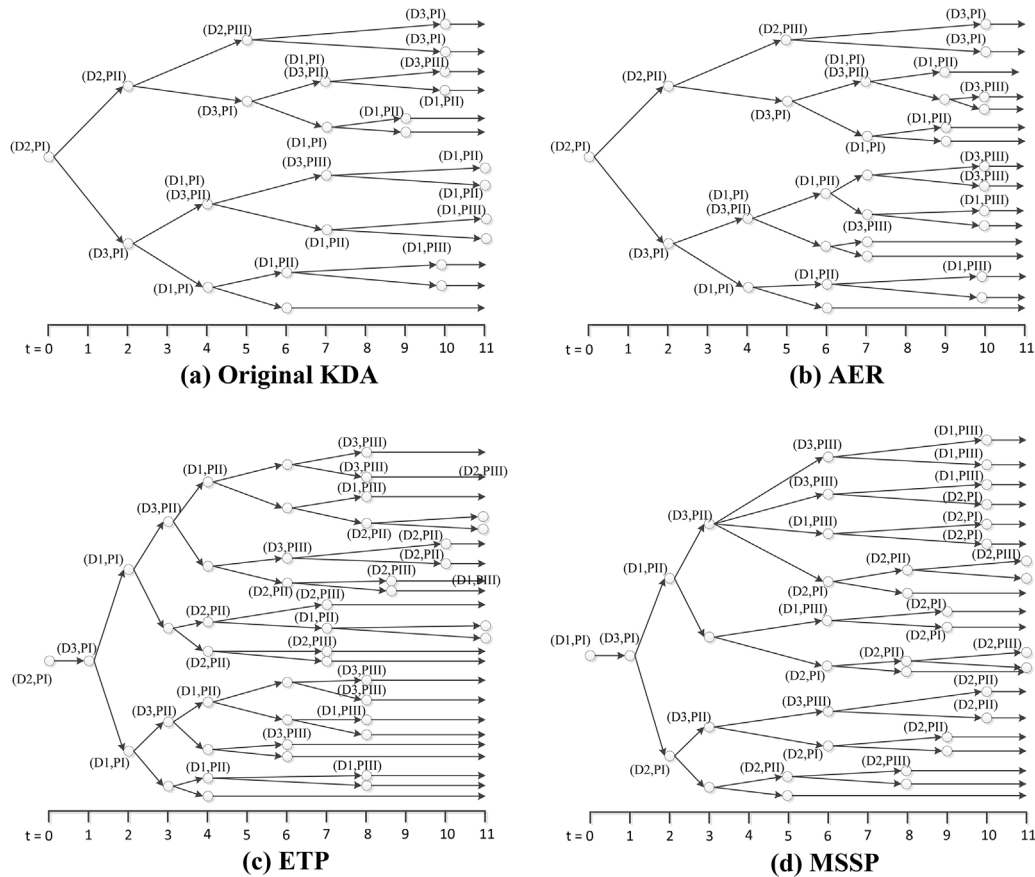
does not have a significant effect on the number of knapsack problems solved. In Fig. 7, this phenomenon is observed with the points that sit on an almost horizontal line at 28 knapsack problems. This horizontal line represents an increase in scenarios without an increase in the number of knapsack problems solved. At 1024 scenarios, Fig. 7 shows that the number of knapsack problems solved increases from 24 to 559 without an increase in the number of scenarios. This occurs when the length of the planning horizon is increased. Increasing the length of the planning horizon neither adds new uncertain parameters nor adds new realizations, thus not impacting the number of scenarios. However, it increases the number knapsack problems solved because there are more decision points.

### 6.3. Impact of the proposed knapsack problem generation rules

The ENPVs and solution times for the KDA using all three knapsack problem generation rules are summarized in Table 2. For completeness, the same information is provided for the solution of the rigorous MSSP. The solution quality does not change significantly among different knapsack problem generation schemes (Table 2). All ENPVs obtained using the KDA approach remain within 5% of the rigorous MSSP ENPVs.

The ENPVs obtained by the ETP and the AER knapsack generation rules are identical for all but the three-product case. The difference in the three-product case is due to the lack of realizations at every time period. The decision trees obtained by the KDA using each of the knapsack problem generation schemes and from the solution of the rigorous MSSP for this case are given in Fig. 8. In the figure, the nodes correspond to decisions. The information in the parentheses represent the (drug, trial) selected at the decision point.

Visually the decision trees obtained using the original KDA (Fig. 8(a)) and the AER decision rule appear (Fig. 8(b)) sparser than the MSSP decision tree (Fig. 8(d)). In contrast, the decision tree for the ETP approach (Fig. 8(c)) is denser than the MSSP decision tree. The ETP approach results in the densest decision tree because it generates new knapsack problems at each time period. The solutions of these knapsack problems, in turn, may result in investment decisions at each time period. For the three-product case, the ETP approach overschedules the trials compared to the original KDA, partly due to how the resource overscheduling constraint functions. This constraint ensures that the total number of resources needed to complete the remaining clinical trials without delay is more than the available number of resources during that same time period. This is implemented using a cumulative approach. For instance,



**Fig. 8.** The decisions trees obtained by KDA for the three product case using (a) the original KDA (Christian and Cremaschi, 2015), (b) the AER, and (c) ETP knapsack generation schemes; and (d) the MSSP (Christian and Cremaschi, 2015) decision tree.

consider a product that requires one resource for the first trial, two resources for the second trial, and three resources for the third trial. The total number of resources needed to complete all three trials is six. The cumulative approach ensures that for the duration of product's clinical trials there are six resources available. It does not account for the fact that the first trial only requires one resource while the third trial requires three. Because the resource requirements for each trial are not differentiated, the algorithm may run into instances where the cumulative number of resources needed is available but the resources needed to start a particular trial at a specific time are not. Solving problems at every time period provides more opportunities for this behavior.

Fig. 8(b) shows the decision tree generated using the AER approach has more decision points than the original KDA decision tree but fewer than the MSSP decision tree. Notice that despite considering new knapsack problems at each realization, the approach fails to obtain the first two decisions correctly, whereas the ETP approach captures this behavior. The AER approach is unable to do so because knapsack problems are only generated after realizations.

The KDA solution times are orders of magnitudes faster than the time required to solve the deterministic equivalent of the rigorous MSSP for all variations of the knapsack sub-problems generation schemes (Table 2). When solution times among different knapsack problem generation schemes are considered, in general, the original KDA yields the solutions the fastest and the KDA using the ETP approach the slowest. The only exception is the solution times for the three-product case in which the times are small enough that the measurement error may exceed the measured time. In the original knapsack algorithm fewer knapsack problems were generated because all realizations are required to occur before new problems

are generated. Using both proposed methods, more knapsack problems are generated. Fig. 9 shows the total number of knapsacks solved using each of the knapsack generation approaches. The problems used are the original problems from Christian and Cremaschi (2015). From our computational studies on problem size, we know that the solution time increases with the number of knapsack problems generated. Notice that for the two-product and three-product cases, the difference between the number of knapsack problems solved in each case is small. In the two-product case, the difference between the number of knapsack problems generated in the original KDA (least) and the ETP approach (most) is only two. In the larger cases (six- and seven-products), the difference in the number of knapsack problems solved increases considerably. In the seven-product case, the difference between the number of knapsack problems generated in the original KDA algorithm (least) and the ETP approach (most) was 462. For all cases, the AER approach generated more knapsack problems than the original KDA but as many or fewer than ETP scheme. In the cases where the realizations for decisions occur at each time period, generating knapsack problems using the ETP scheme is identical to generating knapsack problems using the AER scheme.

#### 6.4. Impact of different formulations of resource overscheduling constraint

The resource overscheduling constraint implemented in the original KDA tries to ensure that there will be resources available to start the subsequent clinical trial if a clinical trial is successful. Using the six problems originally solved in Christian and Cremaschi (2015), we study the impact of different formulations for avoiding

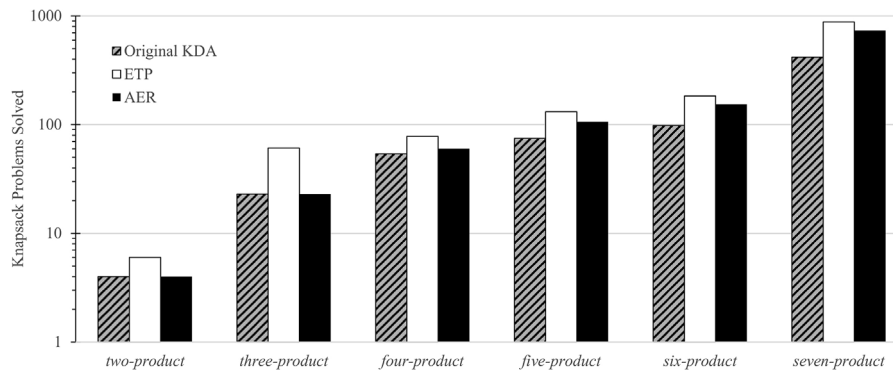


Fig. 9. The number of knapsack problems solved using the original KDA, the ETP, and the AER knapsack generation approaches.

Table 3

The objective functions and solution times of the probabilistic resource constraint and the original resource constraint.

	Original KDA Sub-Problem Generation			
	Probabilistic Resource Overscheduling Constraint		Original Resource Overscheduling Constraint	
	Objective Value	Solve Time (CPU Seconds)	Objective Value	Solve Time (CPU Seconds)
two-product	1097	0.05	1097	0.05
three-product	1165	0.47	1178	0.37
four-product	1665	1.17	1676	0.81
five-product	2038	2.06	2043	1.24
six-product	2399	2.86	2403	1.63
seven-product	2871	17.48	2870	8.17

resource overscheduling: (1) the original resource overscheduling constraint, (2) a penalty term in the objective function, and (3) a probabilistic resource overscheduling constraint.

We varied the penalty coefficient between 1 and 500 and solved each of the base case problems. Varied values of the penalty coefficient resulted in solutions with identical equivalent ENPVs. A closer investigation of the decision trees generated using either value of the penalty coefficient revealed that they did not change for the range considered. This suggests that investments in additional products is significantly limited by the availability of “here-and-now” resources and less impacted by resource availability later in the planning horizon.

The results for the implementation of the probabilistic approach are summarized in Table 3. Comparing the objective function values in Table 3 reveals that values are lower when the probabilistic overscheduling constraint is used in all but the seven product case. In the seven product case, the improved value suggests that the value of allowing additional items to be packed outweighs the losses incurred for products being idle in the pipeline.

The KDA solution times for the proposed approaches to formulating resource overscheduling constraint are similar to that of the original one for the smaller problems (two- and three-product cases). For the larger problems (six- and seven-product cases), there is a noticeable increase in the solution times. For the penalty approach, this increase stems from the increase in the solution times of individual knapsack problems. The time for solving knapsack problems with the penalty term in the objective function was twice as long as solving the original knapsack problems. In each of the case studies, the average time needed to solve a knapsack problem with the penalty term was 0.04 CPU seconds whereas solving the original knapsack problems required 0.02 CPU seconds. In the smaller cases, the number of knapsack problems solved is small, and thus, the difference in the total solution times is insignificant. In larger cases, however, the total number is large, which leads to longer overall solution times for the penalty approach. The difference in the time to solve each knapsack problem with the penalty

term can be attributed to the additional variables that were added to linearize the penalty term in the objective function.

Similar to the penalty approach, the probabilistic approach also shows increased solution time for larger problems (six- and seven-products). The knapsack problems with the probabilistic resource overscheduling constraint use a similar formulation to the original knapsack problems. For smaller problems (two- and three-products), the solution times for the knapsack problems with the probabilistic constraint and the original knapsack problems are identical (0.02 CPU seconds). The larger problems revealed that the knapsack problems with probabilistic constraint take longer to solve (0.03 CPU seconds). The linear relaxation of the knapsack problem with the probabilistic constraint yields a larger feasible region than the original knapsack problem, which yields a looser upper bound. Therefore, for larger instances of the problem with more items that can be packed in a knapsack (i.e., products), it takes longer to solve the problems with the probabilistic constraint.

## 7. Conclusions

The first portion of the presented work studied the sensitivity of the KDA solution to the changes in the values of the fixed parameters (trial cost, revenue, penalty levels, and trial length) of the R&D pipeline management problem. The trial cost, the revenue, and the penalty level parameters were perturbed by  $\pm 10\%$  and  $\pm 25\%$ . The trial length was extended by one and two time periods. In all of the cases, we observed very little change in the decisions obtained by the KDA. When the problem size was increased the accuracy of the KDA solution was not impacted by increasing the number of resources or the length of the planning horizon. However, when the number of trials was increased the KDA was unable to produce a “do-nothing” solution resulting in a large percent gap between the ENPV of the KDA solution and that of the deterministic equivalent of the rigorous MSSP. Increasing the length of the planning horizon caused exponential growth in the solution time. This can be attributed to the increase in the number of knapsack problems solved. Because the KDA only generates knapsack problems to solve

once uncertainty associated with all packed items (i.e., drug-trial pairs) are realized, longer durations reduces the decision points generated by the KDA, and results in sparser decision trees compared to the true solution.

In this work, we considered several modifications to improve the KDA presented in Christian and Cremaschi (2015). The original KDA made decisions after all realizations occurred. Here, we presented two additional approaches, a scheme that generates knapsack problems at every time period and a scheme that generates knapsack problems at each realization. Using a set of 25 problems including the six original problems solved in Christian and Cremaschi (2015), each scheme was tested. The results suggest that generating new problems at each realization provides solutions closer to the deterministic equivalent MSSP than generating problems after all realizations occurred. By solving new knapsack problems at each time period, the solution was often identical to solving new problems at each realization. In the few cases where solving knapsacks at each time period provided a superior solution, the increase in the algorithm time did not justify selection of the ETP approach over the AER approach.

In addition to the sub-problem generation schemas, two additional resource overscheduling constraints in knapsack problems were considered. The two constraints, an objective function penalty approach and a probabilistic constraint approach, yielded solutions

that were worse than the original knapsack approach in all but one case (the largest case with seven products). For this reason, we recommend using the original knapsack resource overscheduling constraint. Future work will examine the use of the KDA as a lower bounding procedure for the solution of the MSSP, and investigate the generalization of the KDA approach to solve problems other than the pharmaceutical R&D pipeline management problem.

**Acknowledgements**

We acknowledge the Auburn University Hopper Cluster for support of this work. The authors greatly appreciate the financial support for this work provided by the US National Science Foundation through the Auburn University Integrative Graduate Research and Education Traineeship (IGERT) program (Award# 1069004), and the NSF Career Grant (Award # 1623417). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation

**Appendix A.**

See Tables A1–A4

**Table A1**  
Parameters for the Two-Product Base Case (2.2.5.2).

Product	Duration		Probability of Success		Cost(\$1 M)		Resource 1 (max = 2)		Resource 2 (max = 3)		rev <sup>max</sup>	γ <sup>L</sup>	γ <sup>D</sup>
	PI	PII	PI	PII	PI	PII	PI	PII	PI	PII			
D1	2	4	0.3	0.5	10	90	1	1	1	2	3100	19.2	44
D2	2	3	0.4	0.6	10	80	1	2	1	1	3250	19.6	56

**Table A2**  
Parameters for the Three-Product Base Case (3.3.12.2).

Product	Duration			Probability of Success			Trial Cost (\$M)			Resource 1 (max = 2)			Resource 1 (max = 3)			rev <sup>max</sup>	γ <sup>L</sup>	γ <sup>D</sup>
	PI	PII	PIII	PI	PII	PIII	PI	PII	PIII	PI	PII	PIII	PI	PII	PIII			
D1	2	4	4	0.3	0.5	0.8	10	90	220	1	1	2	1	2	3	3100	19.2	22
D2	2	3	5	0.4	0.6	0.8	10	80	200	1	2	2	1	1	3	3250	19.6	28
D3	2	3	4	0.3	0.6	0.9	10	90	180	1	1	2	1	1	3	3300	20	26

**Table A3**  
Parameters for the Four-Product Base Case (4.3.6.2).

Product	Duration			Probability of Success			Trial Cost (\$M)			Resource 1 (max = 4)			Resource 2 (max = 3)			rev <sup>max</sup>	γ <sup>L</sup>	γ <sup>D</sup>
	PI	PII	PIII	PI	PII	PIII	PI	PII	PIII	PI	PII	PIII	PI	PII	PIII			
D1	1	1	3	0.3	0.5	0.8	10	90	220	1	1	2	1	2	3	3100	19.2	22
D2	1	2	2	0.4	0.6	0.8	10	80	200	1	2	2	1	1	3	3250	19.6	28
D3	1	1	3	0.3	0.6	0.9	10	90	180	1	1	2	1	1	3	3300	20	26
D4	1	2	2	0.4	0.6	0.8	10	100	170	1	1	2	1	2	3	3000	19.4	24

**Table A4**  
Parameters for the Five-Product Base Case (4.3.6.2).

Product	Duration			Probability of Success			Trial Cost (\$M)			Resource 1 (max = 4)			Resource 2 (max = 3)			rev <sup>max</sup>	γ <sup>L</sup>	γ <sup>D</sup>
	PI	PII	PIII	PI	PII	PIII	PI	PII	PIII	PI	PII	PIII	PI	PII	PIII			
D1	1	1	3	0.3	0.5	0.8	10	90	220	1	1	2	1	2	3	3100	19.2	22
D2	1	2	2	0.4	0.6	0.8	10	80	200	1	2	2	1	1	3	3250	19.6	28
D3	1	1	3	0.3	0.6	0.9	10	90	180	1	1	2	1	1	3	3300	20	26
D4	1	2	2	0.4	0.6	0.8	10	100	170	1	1	2	1	2	3	3000	19.4	24
D5	1	2	3	0.35	0.5	0.9	10	70	210	1	1	2	1	1	3	3150	19.6	24



Table A4 (Continued)

Parameters for the Six-Product Base Case (6.3.6.2)																		
Product	Duration			Probability of Success			Trial Cost (\$M)			Resource 1 (max = 4)			Resource 2 (max = 3)			rev <sup>max</sup>	$\gamma^L$	$\gamma^D$
	PI	PII	PIII	PI	PII	PIII	PI	PII	PIII	PI	PII	PIII	PI	PII	PIII			
D1	1	1	3	0.3	0.5	0.8	10	90	220	1	1	2	1	2	3	3100	19.2	22
D2	1	2	2	0.4	0.6	0.8	10	80	200	1	2	2	1	1	3	3250	19.6	28
D3	1	1	3	0.3	0.6	0.9	10	90	180	1	1	2	1	1	3	3300	20	26
D4	1	2	2	0.4	0.6	0.8	10	100	170	1	1	2	1	2	3	3000	19.4	24
D5	1	2	3	0.35	0.5	0.9	10	70	210	1	1	2	1	1	3	3150	19.6	24
D6	1	2	3	0.45	0.45	0.8	10	85	195	1	2	2	2	1	3	3050	19	25

## Appendix B.

See Tables B1–B3

**Table B1**  
Parameter and Size Perturbation Results for the Two-Product Base Case (OP).

File Name	ENPV	Percent Difference (MSSP)	Percent Difference (OP)	Same as Base KDA solution?	Knapsack Problem Count	KDA Solve Time	MSSP ENPV	MSSP Total Time
2.2.5.3	1097	-0.63	0.00	TRUE	4	0.07	1104	0.15
2.2.5.4	1097	-0.63	0.00	TRUE	4	0.07	1104	0.16
2.2.5.5	1097	-0.63	0.00	TRUE	4	0.07	1104	0.19
2.3.5.2	709	-3.26	-35.36	FALSE	4	0.07	733	0.33
2.4.5.2	523	-5.72	-52.30	FALSE	4	0.07	555	0.58
2.5.5.2	422	-5.03	-61.56	FALSE	5	0.09	444	1.15
2.2.4.2	1101	0.19	0.35	FALSE	4	0.07	1099	0.11
2.2.6.2	1110	0.00	1.17	TRUE	6	0.09	1110	0.15
2.2.7.2	1110	-0.10	1.17	TRUE	6	0.11	1111	0.19
2.2.8.2	1110	-0.39	1.17	TRUE	6	0.11	1114	0.21
2.2.5.2.C.75	1115	-0.52	1.62	TRUE	4	0.08	1121	0.24
2.2.5.2.C.9	1104	-0.59	0.65	TRUE	4	0.07	1111	0.14
2.2.5.2.C1.1	1090	-0.68	-0.65	TRUE	4	0.08	1098	0.13
2.2.5.2.C1.25	1079	-0.75	-1.62	TRUE	4	0.08	1088	0.14
2.2.5.2.R.75	791	-0.88	-27.94	TRUE	4	0.06	798	0.13
2.2.5.2.R.9	975	-0.71	-11.18	TRUE	4	0.08	982	0.16
2.2.5.2.R1.1	1220	-0.57	11.18	TRUE	4	0.07	1227	0.16
2.2.5.2.R1.25	1404	-0.49	27.94	TRUE	4	0.08	1411	0.15
2.2.5.2.GL.75	1110	-0.75	1.14	TRUE	4	0.08	1118	0.15
2.2.5.2.GL.9	1102	-0.68	0.46	TRUE	4	0.08	1110	0.15
2.2.5.2.GL1.1	1092	-0.59	-0.46	TRUE	4	0.07	1099	0.15
2.2.5.2.GL1.25	1085	-0.52	-1.14	TRUE	4	0.08	1090	0.14
2.2.5.2.GD.75	1099	-0.45	0.18	TRUE	4	0.16	1104	0.15
2.2.5.2.GD.9	1098	-0.56	0.07	TRUE	4	0.08	1104	0.15
2.2.5.2.GD1.1	1096	-0.71	-0.07	TRUE	4	0.08	1104	0.14
2.2.5.2.GD1.25	1095	-0.81	-0.18	TRUE	4	0.07	1104	0.15
2.2.5.2.TD1	1087	0.87	-0.89	FALSE	4	0.03	1078	0.14
2.2.5.2.TD2	1077	0.72	-1.81	FALSE	4	0.03	1070	0.14
2.2.5.2.PC60	1097	-0.63	0.00	TRUE	4	0.08	1104	0.15
2.2.5.2.PC30	1097	-0.63	0.00	TRUE	4	0.08	1104	0.15
2.2.5.2.PC0	1121	0.00	2.13	FALSE	4	0.08	1121	0.15
Average		-0.84						

**Table B2**

Parameter and Size Perturbation Results for the Three-Product Base Case (OP).

File Name	ENPV	Percent Difference (MSSP)	Percent Difference (OP)	Same as Base KDA solution?	Knapsack Problem Count	KDA Solve Time	MSSP ENPV	MSSP Total Time
3.3.12.3	1180	-0.67	0.17	FALSE	21	0.46	1188	5.44
3.3.12.4	1180	-0.67	0.17	FALSE	21	0.43	1188	5.82
3.3.12.5	1180	-0.79	0.17	FALSE	21	0.48	1189	5.88
3.4.12.2	761	-1.60	-35.38	FALSE	21	0.75	773	16.44
3.5.12.2	417	-15.29	-64.60	FALSE	21	0.51	492	44.90
3.6.12.2	240	-22.10	-79.64	FALSE	31	0.94	308	109.61
3.3.11.2	1177	-0.92	-0.04	FALSE	20	0.78	1188	4.68
3.3.13.2	1174	-1.24	-0.31	FALSE	27	1.07	1189	5.68
3.3.14.2	1178	-1.25	0.02	FALSE	28	1.10	1193	10.16
3.3.15.2	1177	-1.68	-0.10	FALSE	33	1.41	1197	6.46
3.3.12.2.C.75	1225	-0.92	4.03	FALSE	23	0.61	1237	5.39
3.3.12.2.C.9	1197	-0.96	1.61	FALSE	23	0.58	1208	5.46
3.3.12.2.C1.1	1159	-1.01	-1.61	FALSE	23	0.58	1171	5.32
3.3.12.2.C1.25	1130	-1.00	-4.03	FALSE	23	0.58	1142	5.35
3.3.12.2.R.75	804	-1.11	-31.76	FALSE	23	0.61	813	5.34
3.3.12.2.R.9	1028	-1.03	-12.71	FALSE	23	0.55	1039	5.25
3.3.12.2.R1.1	1327	-1.00	12.71	FALSE	23	0.57	1341	5.40
3.3.12.2.R1.25	1552	-0.92	31.79	FALSE	23	0.56	1567	5.20
3.3.12.2.GL.75	1209	-0.96	2.67	FALSE	23	0.59	1221	5.28
3.3.12.2.GL.9	1190	-0.99	1.07	FALSE	23	0.60	1202	5.36
3.3.12.2.GL1.1	1165	-1.04	-1.07	FALSE	23	0.60	1177	5.24
3.3.12.2.GL1.25	1146	-1.11	-2.67	FALSE	23	0.55	1159	5.42
3.3.12.2.GD.75	1179	-1.02	0.08	FALSE	23	0.57	1191	5.28
3.3.12.2.GD.9	1178	-1.02	0.03	FALSE	23	0.57	1190	5.28
3.3.12.2.GD1.1	1177	-1.02	-0.03	FALSE	23	0.58	1189	5.40
3.3.12.2.GD1.25	1177	-1.02	-0.08	FALSE	23	0.58	1189	5.28
3.3.12.2.TD1	1132	-1.85	-3.95	FALSE	13	0.31	1153	5.60
3.3.12.2.TD2	1083	-2.39	-8.09	FALSE	7	0.28	1109	5.66
3.3.12.2.PC60	1183	-1.61	0.48	FALSE	33	0.82	1203	5.28
3.3.12.2.PC30	1213	-0.58	2.96	FALSE	31	0.77	1220	5.37
3.3.12.2.PC0	1214	-0.59	3.09	FALSE	27	0.67	1221	5.27
Average		-2.24						

**Table B3**

Parameter and Size Perturbation Results for the Four-Product Base Case (OP).

File Name	ENPV	Percent Difference (MSSP)	Percent Difference (OP)	Same as Base KDA solution?	Knapsack Problem Count	KDA Solve Time	MSSP ENPV	MSSP Total Time
4.3.6.3	1619	-2.63	-3.39	FALSE	63	1.65	1663	22.76
4.3.6.4	1619	-2.25	-3.39	FALSE	63	1.90	1656	23.58
4.3.6.5	1624	-1.67	-3.09	FALSE	49	1.43	1651	24.40
4.4.6.2	1095	-3.64	-34.65	FALSE	50	1.42	1136	126.59
4.5.6.2	687	-13.17	-58.99	FALSE	24	0.93	791	554.35
4.6.6.2	405	-22.59	-75.80	FALSE	24	1.35	524	2067.58
4.3.5.2	1664	-0.73	-0.67	FALSE	32	2.30	1677	17.83
4.3.7.2	1687	-0.01	0.67	FALSE	80	5.46	1687	25.58
4.3.8.2	1695	0.10	1.15	FALSE	111	7.74	1693	29.49
4.3.9.2	1699	-0.01	1.39	FALSE	139	9.73	1699	59.11
4.3.6.2.C.75	1747	-0.66	4.24	FALSE	54	3.20	1758	24.45
4.3.6.2.C.9	1704	-0.67	1.70	FALSE	54	3.13	1716	23.45
4.3.6.2.C1.1	1647	-0.37	-1.70	FALSE	54	2.84	1653	24.05
4.3.6.2.C1.25	1605	-0.54	-4.24	FALSE	54	2.18	1613	23.53
4.3.6.2.R.75	1160	-0.86	-30.75	FALSE	54	2.14	1170	23.48
4.3.6.2.R.9	1469	-1.07	-12.30	FALSE	54	2.06	1485	23.48
4.3.6.2.R1.1	1882	-0.59	12.30	FALSE	54	2.06	1893	23.55
4.3.6.2.R1.25	2191	-0.70	30.76	FALSE	54	2.05	2206	23.29
4.3.6.2.GL.75	1699	-0.84	1.42	FALSE	54	2.07	1714	23.46
4.3.6.2.GL.9	1685	-0.54	0.57	FALSE	54	2.02	1694	23.70
4.3.6.2.GL1.1	1666	-0.77	-0.57	FALSE	54	2.04	1679	23.62
4.3.6.2.GL1.25	1652	-0.79	-1.42	FALSE	54	2.04	1665	23.46
4.3.6.2.GD.75	1677	-0.68	0.10	FALSE	54	1.98	1689	23.55
4.3.6.2.GD.9	1676	-0.78	0.04	FALSE	54	1.99	1689	23.38
4.3.6.2.GD1.1	1675	-0.51	-0.04	FALSE	54	1.91	1684	23.63
4.3.6.2.GD1.25	1674	-0.56	-0.10	FALSE	54	1.88	1683	23.48
4.3.6.2.TD1	1602	-2.04	-4.40	FALSE	19	0.21	1636	24.09
4.3.6.2.TD2	1486	-1.55	-11.32	FALSE	18	0.21	1510	23.86
4.3.6.2.PC60	1676	-1.57	0.02	FALSE	85	3.10	1703	23.66
4.3.6.2.PC30	1706	-0.74	1.79	FALSE	81	2.97	1718	23.23
4.3.6.2.PC0	1714	-0.40	2.31	FALSE	81	3.02	1721	23.46
Average		-2.06						

Table B3 (Continued)

Parameter and Size Perturbation Results for the Five-Product Base Case (OP)								
File Name	ENPV	Percent Difference (MSSP)	Percent Difference (OP)	Same as Base KDA solution?	Knapsack Problem Count	KDA Solve Time	MSSP ENPV	MSSP Total Time
5.3.6.3	1982	-2.82	-2.95	FALSE	90	1.96	2040	286.94
5.3.6.4	1982	-2.62	-2.95	FALSE	90	3.81	2036	294.54
5.3.6.5	1994	-1.90	-2.35	FALSE	87	4.11	2033	296.37
5.4.6.2	1323	-3.77	-35.24	FALSE	68	3.35	1375	2963.13
5.5.6.2	860	-10.83	-57.91	FALSE	26	2.28	964	21376.09
5.6.6.2	509	-18.86	-75.08	FALSE	26	0.61	627	112356.45
5.3.5.2	2038	0.11	-0.25	FALSE	51	1.10	2035	227.08
5.3.7.2	2059	-0.10	0.79	FALSE	147	3.10	2061	321.21
5.3.8.2	2064	0.03	1.05	FALSE	194	4.15	2063	366.56
5.3.9.2	2084	0.44	2.01	FALSE	293	6.32	2075	414.11
5.3.6.2.C.75	2128	-0.28	4.18	FALSE	75	1.59	2134	278.36
5.3.6.2.C.9	2077	-0.25	1.67	FALSE	75	1.60	2082	278.04
5.3.6.2.C1.1	2008	-0.50	-1.67	FALSE	75	1.60	2019	851.48
5.3.6.2.C1.25	1957	-0.54	-4.18	FALSE	75	1.55	1968	313.37
5.3.6.2.R.75	1413	-0.30	-30.80	FALSE	75	1.61	1418	288.00
5.3.6.2.R.9	1791	-0.41	-12.32	FALSE	75	1.67	1798	280.77
5.3.6.2.R1.1	2294	-0.15	12.32	FALSE	75	1.62	2298	280.53
5.3.6.2.R1.25	2672	-0.66	30.82	FALSE	75	1.62	2690	279.43
5.3.6.2.GL.75	2074	-0.47	1.53	FALSE	75	1.61	2084	279.67
5.3.6.2.GL.9	2055	-0.31	0.61	FALSE	75	1.62	2061	284.72
5.3.6.2.GL1.1	2030	-0.77	-0.61	FALSE	75	1.63	2046	282.60
5.3.6.2.GL1.25	2011	-0.65	-1.53	FALSE	75	1.65	2025	283.77
5.3.6.2.GD.75	2045	-1.20	0.11	FALSE	75	1.58	2069	280.57
5.3.6.2.GD.9	2043	-0.84	0.04	FALSE	75	1.56	2061	287.98
5.3.6.2.GD1.1	2042	-0.19	-0.04	FALSE	75	1.57	2046	280.94
5.3.6.2.GD1.25	2040	-0.37	-0.11	FALSE	75	1.65	2048	285.80
5.3.6.2.TD1	1957	-2.32	-4.20	FALSE	21	0.12	2004	279.89
5.3.6.2.TD2	1840	-1.22	-9.96	FALSE	5	0.25	1862	286.54
5.3.6.2.PC60	2097	-1.10	2.68	FALSE	215	9.72	2121	282.39
5.3.6.2.PC30	2118	-0.45	3.68	FALSE	243	11.21	2127	280.09
5.3.6.2.PC0	2120	-0.39	3.78	FALSE	243	11.28	2128	282.89
Average		-1.73						

## References

- Christian, B., Cremaschi, S., 2015. Heuristic solution approaches to the pharmaceutical R&D pipeline management problem. *Comput. Chem. Eng.* 74, 34–47.
- Colvin, M., Maravelias, C.T., 2008. A stochastic programming approach for clinical trial planning in new drug development. *Comput. Chem. Eng.* 32, 2626–2642, <http://dx.doi.org/10.1016/j.compchemeng.2007.11.010>.
- Colvin, M., Maravelias, C.T., 2009. Scheduling of testing tasks and resource planning in new product development using stochastic programming. *Computers & Chemical Engineering* 33, 964–976, <http://dx.doi.org/10.1016/j.compchemeng.2008.09.010>.
- Goel, V., Grossmann, I.E., 2004. A stochastic programming approach to planning of offshore gas field developments under uncertainty in reserves. *Comput. Chem. Eng.* 28 (8), 1409–1429, <http://dx.doi.org/10.1016/j.compchemeng.2003.10.005>.
- Goel, V., Grossmann, I.E., 2006. A class of stochastic programs with decision dependent uncertainty. *Math. Program.* 108 (2), 355–394.
- Gupta, V., Grossmann, I.E., 2014. Multistage stochastic programming approach for offshore oilfield infrastructure planning under production sharing agreements and endogenous uncertainties. *Journal of Petroleum Science and Engineering* 124, 180–197, <http://dx.doi.org/10.1016/j.petrol.2014.10.006>.
- Mercier, L., Hentenryck, P. Van, 2008. *Amsaa: A Multistep Anticipatory Algorithm for Online Stochastic Combinatorial Optimization.*, pp. 173–187.
- Mercier, L., 2009. *AMSAA: A Multistep Anticipatory Algorithm for Online Stochastic Combinatorial Optimization.*
- Solak, S., Clarke, J.P.B., Johnson, E.L., Barnes, E.R., 2010. Optimization of R&D project portfolios under endogenous uncertainty. *Eur. J. Oper. Res.* 207 (1), 420–433, <http://dx.doi.org/10.1016/j.ejor.2010.04.032>.
- Subramanian, D., Pekny, J.F., Reklaitis, G.V., 2001. A simulation-optimization framework for research and development pipeline management. *AIChE J.* 47 (10), 2226–2242, <http://dx.doi.org/10.1002/aic.690471010>.
- Tarhan, B., Grossmann, I.E., Goel, V., 2008. *A Multistage Stochastic Programming Approach for the Planning of Offshore Oil or Gas Field Infrastructure Under Decision Dependent Uncertainty.*, pp. 1–61.
- Tarhan, B., Grossmann, I.E., Goel, V., 2013. Computational strategies for non-convex multistage MINLP models with decision-dependent uncertainty and gradual uncertainty resolution. *Annals of Operations Research* 203 (1), 141–166, <http://dx.doi.org/10.1007/s10479-011-0855-x>.