



# A new Lagrangian solution scheme for non-decomposable multidisciplinary design optimization problems

Bayan Hamdan<sup>1</sup> · Pingfeng Wang<sup>1</sup>

Received: 2 December 2022 / Revised: 23 March 2023 / Accepted: 24 May 2023 / Published online: 29 June 2023  
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

## Abstract

Multidisciplinary optimization problems exist in many disciplines and constitute a large percentage of problems in industry. Due to their wide-scale applicability, significant research efforts have been spent on developing effective methods that can not only derive accurate solutions but also improve the computational efficiency in the problem solving process. As a result, different algorithms are proposed to coordinate the solution of the different disciplines. However, in realistic problems, the coupling across these disciplines introduces difficulty in the coordination between them. In this paper, a new hybrid meta-heuristic method based on a Lagrangian relaxation of complicating constraints is introduced to reduce the coupling between disciplines in such systems. The proposed scheme identifies complicating constraints and implements a Lagrangian relaxation scheme that allows the constraint to be decomposed over different subproblems. This reduces the coupling across the disciplines and improves the coordination between them. The developed algorithm has been tested on numerical case studies as well as an engineering problem to demonstrate its efficacy as compared with existing methods in the literature for multidisciplinary optimization problems with strong links between subproblems as well as the scalability.

**Keywords** Multidisciplinary design optimization · Coupled systems · Lagrangian relaxation · Analytical target cascading

## 1 Introduction

Within the complex nature of physical, social and technical systems, their exist interconnected subsystems whose interactions need to be accounted for when making system-wide decisions. This allows the decision maker to make globally optimal decisions for the overall system. However, the complex nature of these systems introduces a computational burden that may be too large to justify system-wide optimization, where the problem is solved All-At-Once (AAO). Due to this widespread problem, many algorithms try to address these specific problem structures that consist or inter-connected disciplines with shared variables and parameters. Such problems are referred to as Multidisciplinary Optimization (MDO) problems which are tackled by decomposing the different disciplines into

subproblems in distributed methods, or through monolithic formulations Tedford and Martins (2010). Monolithic formulations include approaches that have a single system-wide optimization, where the decomposed system interactions are determined in a single iteration. Monolithic formulations, such as multidisciplinary design feasible formulations, tend to be easy to implement and solve, but their solutions could scale poorly with the complexity of the problem Tedford and Martins (2010) and this motivates new studies that use Machine Learning methods for solving MDO problems Ramu et al. (2022). Similarly, for staged problems, the different stages could be treated as disciplines, however, their is tight coupling across the stages that needs to be accounted for Hamdan et al. (2019). On the other hand, the performance of decomposition schemes is dependent on the complexity of the subproblems as well as the connections between them Santiago et al. (2014). The term complexity, here, refers to the number of constraints and the dimensionality and number of variables in the problem that could make solving each discipline difficult, in addition to the number of constraints and variables that link the different disciplines together.

Decomposition-based methods use iterative solution schemes to coordinate between the different subproblems.

---

Responsible Editor: Seonho Cho

✉ Pingfeng Wang  
pingfeng@illinois.edu

<sup>1</sup> Department of Industrial and Systems Engineering,  
University of Illinois Urbana-Champaign, 104 S Mathews  
Ave, Urbana, Illinois 61801, USA

This coordination strategy allows the discrepancies between the different subproblems to be minimized with each iteration, approaching a global optimal solution to the problem. The behavior of such methods is highly dependent on the interactions between the subproblems Martins and Kennedy (2021). Therefore, such methods are ideal for problems where the subproblems are modular and have little connections between the other subproblems. When it comes to real-life systems, these links are not always trivial and could cause iterative schemes to diverge or result in a far from optimal solution Lizier et al. (2018). To tackle these complex system structures, the links between the subproblems are usually oversimplified from the true system to facilitate solving the problem. This can make the problem solution not representative of reality or could even result in solutions that are not feasible to the real problem. In order to handle the coordination of MDO problems with complicating variables and constraints, the links across the subproblems need to be reduced to facilitate their coordination with existing MDO methods. As the dimensionality of the problem increases, the links become more intricate. Therefore, decomposing the links across disciplines greatly facilitates their solution, especially when considering iterative MDO solution approaches.

In this paper, a decomposition-based solution scheme is proposed to tackle complex dependencies between modular systems for non-decomposable multidisciplinary optimization problems, while maintaining the quality of the solutions and mitigating the solution complexity. We present a nested Analytical Target Cascading - Lagrangian Relaxation (LR-ATC) solution scheme which can relax complicated links between subproblems to enable tractable solutions. The proposed method can be applied to problems where the relationship between subproblems is complex and can therefore result in improved system-wide optimization for modular problems. Numerical case studies are implemented to measure the effectiveness of the proposed approach over benchmark methods in the literature as well as study the scalability of the proposed solution scheme. A physical system case study is also implemented to highlight the applicability of the proposed method to real case studies.

The rest of the paper is organized as follows. Section 2 introduces the relevant methods in the literature. Section 3 presents the proposed solution scheme. Section 4 presents two numerical case studies and details how the proposed scheme was implemented to solve the problem and how well it scales with the size of the problem. It also highlights the benefit of the proposed method when it comes to the quality of the solution as well as the computational time. Section 5 presents the Golinski Speed Reducer case study and showcases the applicability of the ATC-LR method to real system applications as well as highlights the benefit of the method. Lastly, Sect. 6 concludes the paper.

## 2 Relevant literature

In this section, some of the most prominent solution methods for coupled systems are reviewed. A comparison between the different methods is drawn and the benefit and applicability of each method is highlighted.

Analytical Target Cascading (ATC) is a hierarchical decomposition method for solving problems with coupled systems Kim et al. (2003). The method was first proposed by Kim et al. (2003) to create a tool to propagate top level targets down in a hierarchical scheme. Since then, many studies formalize this solution scheme in a framework and integrate it with lagrangian methods to improve its applicability to complex problems Kim et al. (2006), Kim et al. (2003), Tosserams et al. (2010), Kang et al. (2014). ATC is a framework that decomposes formulation through hierarchical levels where the top level contains the system targets and the bottom levels try to match the targets from above. Additionally, there is target matching across different submodels within the same level that have shared variables. ATC is an iterative scheme where the exterior objective is to minimize the discrepancies between the targets. The method is generally applied in three stages: dividing the formulation into coupled sub-problems (in the case of clearly coupled systems, identification of each subsystem and linking variables), setting the targets for each subsystem depending on system links and solving the sub-problems in a coordinated scheme Kim et al. (2003).

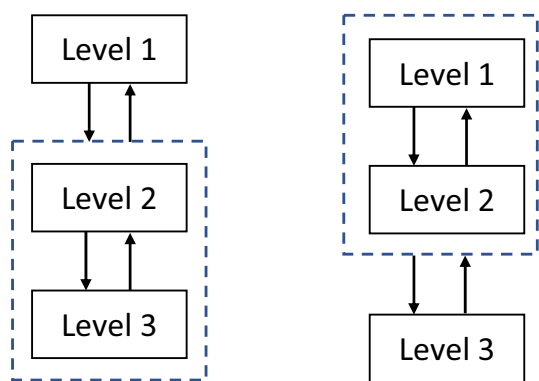
Lagrangian Relaxation (LR) is an exterior penalty method which approximates difficult problems with simpler problems in order to facilitate solving complicated problems. LR is implemented by identifying (a set of) complicating constraints and removing them from the constraint space. However, the model penalizes violation of these constraints through including a penalty term in the objective function. This penalty term usually consists of the constraint violation multiplied by a penalty parameter. LR is then usually solved in an iterative solution scheme where the value of the penalty parameter is optimized Fisher (1985). Different ways of updating the penalty parameter exist (subgradient method, ect.), however, in most cases setting a high penalty parameter value tends to improve the quality of the solution. Nonetheless, a trade-off exists between solution time and the quality of the solution when it comes to a choice of a penalty parameter. That is why iterative solution schemes to select the best penalty parameter values exist Hamdan and Diabat (2020). Eqn 1 shows the general form of the relaxed problem given the original objective,  $c^T x$ , the original constraint  $Ax \leq B$  and the penalty parameter  $\lambda$ .

$$\text{Maximize}_{\lambda} \quad \min_x c^T x + \lambda(Ax - B) \quad (1)$$

Augmented Lagrangian (AL) method is an extension to the LR method where the penalty term is augmented with an additional term which is added to avoid ill-conditioning of the problem Afonso et al. (2010). This parameter is a quadratic penalty function that also serves as a convexifying term. The main benefit of this method is that the penalty parameter does not need to be too large in order to obtain an optimal solution to the original problem Papalambros and Wilde (2000). The updated form of the objective is now shown in Eqn. 2. In this equation,  $\mu$  is an additional penalty parameter for the augmenting term.

$$\text{Maximize}_{\lambda} \quad \min_x c^T x + \lambda(Ax - B) + \frac{\mu}{2} \|Ax - B\|^2 \quad (2)$$

The method has been extended further by using the Alternating Direction Method of Multipliers (ADMM) to reduce the cost incurred with coordination among different levels of the decomposed problem Tosserams et al. (2006). The updated method is referred to as AL-AD ATC which represents an Augmented Lagrangian ATC approach with an Alternating Direction Method of Multipliers. ADMM is an extension to the Method of Multipliers (MM) where instead of iteratively solving an inner loop problem as shown in Fig. 1. The ADMM takes into consideration that in a hierarchical scheme, the odd levels and even levels can each be solved in parallel since odd levels only need responses from even levels and vice versa. Additionally, since in ATC, models along the same level can be solved in parallel, the ADMM method can drastically reduce computational time by parallelizing model optimization along a single level and throughout the levels Tosserams et al. (2005). It also solves each inner loop once instead of the iterative inner loop structure that is very computationally expensive. The results of ADMM are also shown to converge to the results of the original ATC formulation for certain conditions on the problem structure.



a. Nested bottom-up      b. Nested top-down

Fig. 1 Inner loop coordination strategy for the method of multipliers

### 3 A new Lagrangian solution scheme

This section details the developed new Lagrangian solution scheme for the multidisciplinary optimization problems with non-decomposable subsystems. Section 3.1 explains the non-decomposable problem structures, and Sect. 3.2 then details the developed the LR-AL ATC algorithms for the non-decomposable MDO problems. The motivation for the proposed algorithm is that it can handle non-decomposable problem structures with strong links across subproblems. This is done by relaxing the complicating constraints and solving a meta-heuristic, iterative algorithm that can update the relaxation penalties while coordinating the different subproblems.

#### 3.1 Non-decomposable problem structures

In this section we define non-decomposable subsystems. This work targets systems that have an embedded grouping of subsystems that are tightly connected. This hinders the ability of the model to be decomposed into subsystems which can greatly increase the computational time of large and complex problems. In order to illustrate the class of problems targeted in this work, a simple mathematical example is presented in Eqns. 3a to 3g. In this problem, there are six local decision variables represented by  $x_i$ , two partially shared variables across two subproblems,  $u_i$ , and a shared variable across all subproblems,  $v$ . Although this simple mathematical model has an inherent subgrouping of its variables that make it a strong candidate for solving the problem through decomposition, the existence of constraint Eqn. 3g makes the problem non-decomposable. Figure 2 illustrated the decomposition

	$x_1$	$x_2$	$u_1$	$x_3$	$x_4$	$u_2$	$x_5$	$x_6$	$v$
$C_1$	x	x		Subgroup 1					
$C_2$	x	x	x						
$C_3$				x	x				
$C_4$				x	Subgroup 2		x	x	
$C_5$						x	Subgroup 3		
$C_6$	x			x			x	x	x

Fig. 2 Decomposed outline of the subgroups found in the model structure highlighting the non-decomposable nature of the problem. Shared variables across subgroups are highlighted in gray

matrix of the problem. A clear decomposition structure exists for the local variables and partially shared variables, yet, the shared variable found in Eqn. 3g creates strong links along all three subproblems. The gray shaded areas in the figure represent the shared variables. Current decomposition methods have been shown to be effective with partially shared variables (highlighted in light gray), however, strong links between the subproblems (highlighted in dark gray) greatly hinder the effectiveness of decomposition schemes.

$$\min f = x_2^2 + 5x_6 \tag{3a}$$

$$\text{s.t. } x_1^2 + 3x_2 + 2 \leq 0 \tag{3b}$$

$$x_1 - u_1 + x_2 + 2 \leq 0 \tag{3c}$$

$$-x_3 + 3x_4 - 4 \leq 0 \tag{3d}$$

$$-u_1 + x_5 - u_2 + 2 \leq 0 \tag{3e}$$

$$2x_6 - u_2 + 3 \leq 0 \tag{3f}$$

$$x_2 + vx_6 - x_4 + 2 \leq 0 \tag{3g}$$

$$x_i; \quad u_i \geq 0, \quad \forall i \tag{3h}$$

$$v \geq 0$$

### 3.2 The LR-AL ATC methodology

In this section, a new Lagrangian Relaxation solution scheme is introduced, which can be used to solve analytical mathematical models that have a non-decomposable or slightly decomposable structures. In order to illustrate how the algorithm can be applied to a general set of problems, general variable and constraint sets will be used. The general form of the problem being studied is presented in Eqn 4. In this formulation,  $x$  represents the vector of decision variables, indexed by  $i$ , where the full set of decision variables is given by  $x = [x_1^T, x_2^T, \dots, x_n^T]^T$ .  $c^T$  represents the cost vector indexed by  $i$ . The objective function is to minimize the total cost of the chosen decision, given by the product  $C^T x$ . The matrix  $A$  represents the left hand side of the constraints and  $B$  represents the vector of the right hand side of the constraints.

$$\begin{aligned} &\text{Minimize } c^T x \\ &\text{s.t. } Ax \leq B \end{aligned} \tag{4}$$

The proposed method combines an AL-AD ATC solution scheme in an iterative Lagrangian Relaxation solution scheme. This approach can handle the added complexity of complicating constraints that make ATC inapplicable for applications where the system is not perfectly modular, or where the links between the subsystems are complex. LR can be used to relax the complicating constraints that link the subsystems in order to improve the ability of ATC to coordinate between the subproblems. Figure 3 shows how the simple mathematical example presented in Eqns. 3 can be converted into a decomposable structure through relaxing the coupling constraint. The constraint causing the strong linkage between subproblems, highlighted in Fig. 2, is

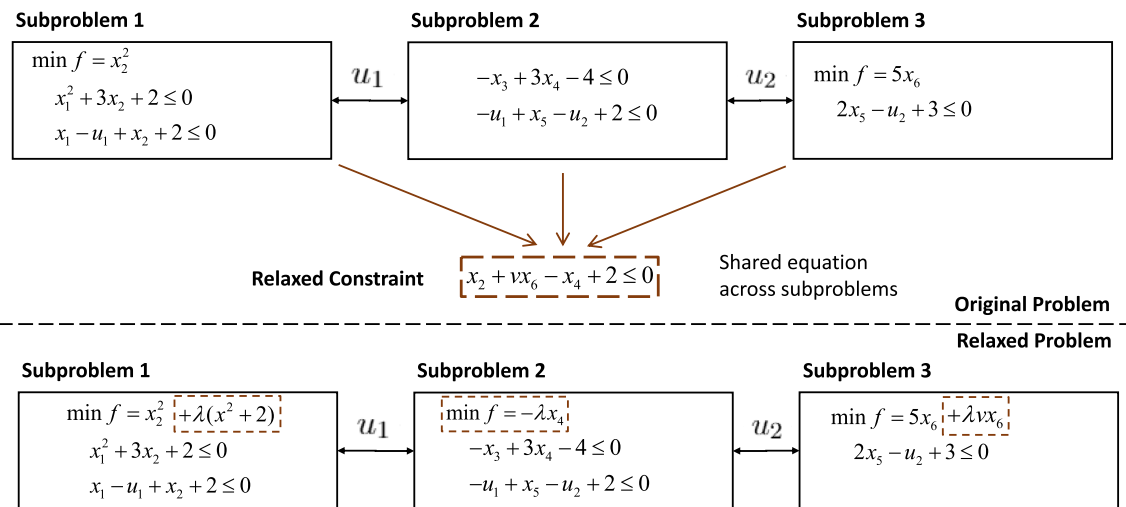


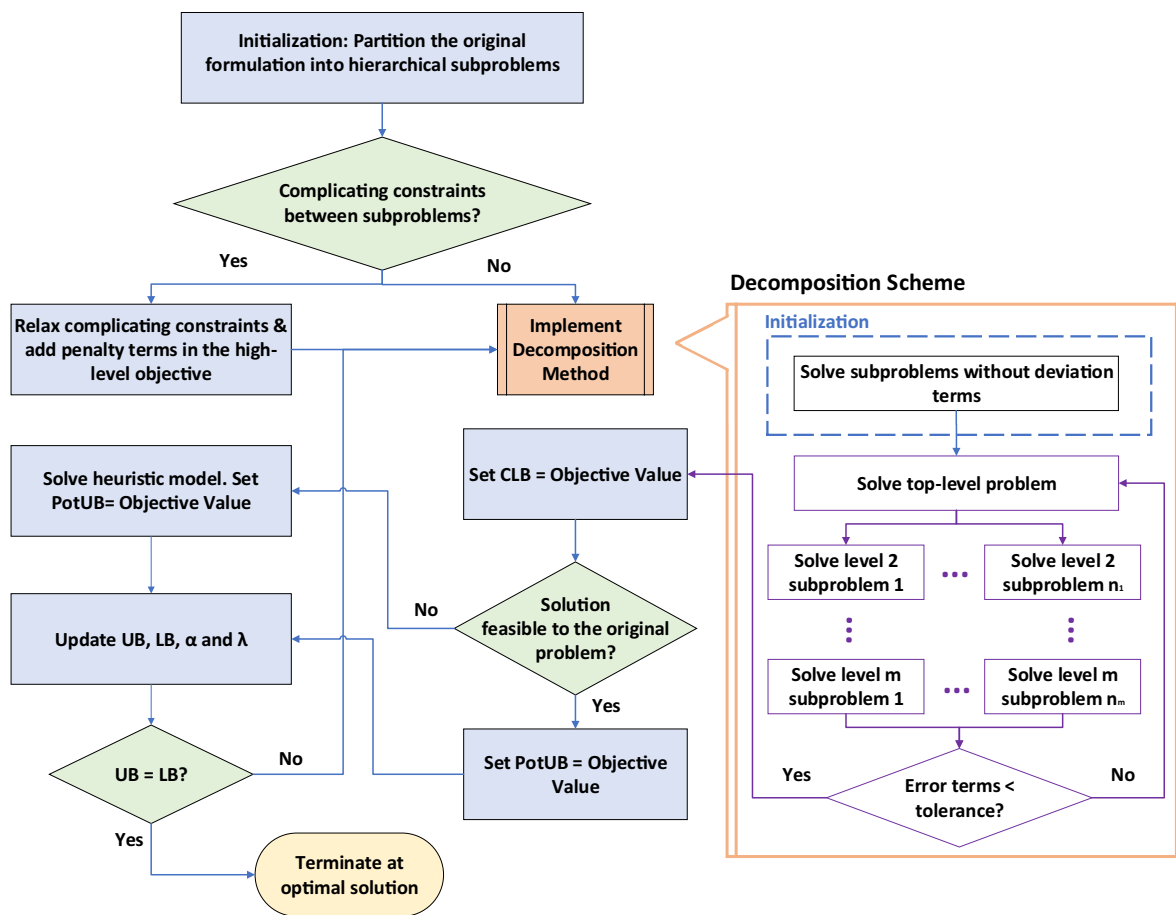
Fig. 3 Benefit of relaxing a single complicating constraint for non-decomposable systems

relaxed. Due to the additive property of the objective function, the relaxed constraint is distributed over the subproblems in their respective objective functions.

The proposed solution scheme is shown in Fig 4, where the structure is outlined for a general implementation of the solution algorithm. The pseudocode presented in Fig 5 presents the proposed nested solution framework. The overall solution scheme is presented in Fig 5 where the outer loop is defined for a general problem structure. The inputs to the algorithm include the Cost vector for the objective function,  $C$ , the matrix,  $A$ , which represents the left-hand side coefficients for the constraints, and the vector,  $B$ , which represents the right hand side constants for the constraints. For this case, the relaxed set of constraints is represented by inequality constraints, however, for equality constraints, this can be easily modified. In the case of equality constraints, the absolute value of the difference  $Ax - B$  is used to determine the stepsize. The Lower Bound ( $LB$ ) and the Upper Bound

( $UB$ ) are calculated based on the values of the objectives of the Relaxed MDO solution and the heuristic solution. For a minimization problem, the  $LB$  value is the optimal but not necessarily feasible solution obtained from the MDO formulation. The  $UB$ , on the other hand, represents the feasible, yet not necessarily optimal solution obtained from the heuristic model. Moreover, the pseudocode shows the case for a minimization problem, in the case of a maximization objective, the  $UB$  and  $LB$  values can be reversed. The parameters for the algorithm include the stepsize,  $\alpha$ , the penalty parameter (this can be a vector in the case of relaxing a set of constraints),  $\lambda$ , and the number of consecutive iterations with no improvement in the value of the upper bound (for a minimization problem),  $noimp$ .

The algorithm then enters a loop until the upper bound and lower bound converge within a predefined tolerance,  $\tau_1$  or the scale factor,  $\epsilon$  falls below a tolerance,  $\tau_2$ . Inside the outer loop, the Relaxed Master Problem (RMP) is solved



$UB$  = upper bound;  $LB$  = lower bound;  $\alpha$  = stepsize;  $\lambda$  = penalty parameter;  $PotUB$  = potential upper bound;  $CLB$  = current lower bound

Fig. 4 Proposed LR-AL ATC Method Framework

**Algorithm 1:** LR-ATC pseudocode

```

Data: Matrices:  $\mathbf{A}$  ; Vectors:  $\mathbf{C}, \mathbf{B}$ 
Result: Solution  $(\mathbf{x}, \lambda)$  to RMP
begin
   $\varepsilon \leftarrow 2$ ;  $\lambda \leftarrow 1$ ;  $noimp \leftarrow 0$ ;
  repeat
    Solve RelaxedMasterProblem using ATC Subroutine;
     $CLB \leftarrow COFV(RelaxedMasterProblem)$ ;
     $check1 \leftarrow 0$ ;
    if  $\mathbf{Ax} > \mathbf{B}$  then
      |  $check1 \leftarrow check1 + 1$ ;
    end
    if  $check1 == 0$  then
      |  $PotUB \leftarrow COFV(RelaxedMasterProblem)$ ;
    else
      | Run the Heuristic Subroutine
      |  $PotUB \leftarrow COFV(HeuristicModel)$ ;
    end
    if  $UB > PotUB$  then
      |  $UB \leftarrow PotUB$ ;
    else
      |  $noimp \leftarrow noimp + 1$ ;
    end
    if  $\mathbf{Ax} \leq \mathbf{B}$  then
      |  $\alpha \leftarrow 0$ ;
    else
      |  $\alpha \leftarrow \frac{\varepsilon(UB-CLB)}{\sqrt{(\mathbf{Ax}-\mathbf{B})^2}}$ ;
      | if  $\alpha \neq 0$  then
      | |  $\lambda \leftarrow \max\{0, \lambda + \alpha \times (\mathbf{Ax} - \mathbf{B})\}$ ;
      | end
      | if  $CLB > LB$  then
      | |  $LB \leftarrow CLB$ ;
      | end
      | if  $noimp > 5$  then
      | |  $\varepsilon \leftarrow \varepsilon/2$ ;
      | |  $noimp \leftarrow 0$ ;
      | end
    end
  end
until  $(|UB - LB| < \tau_1)$  or  $(\varepsilon < \tau_2)$ ;
end

```

**Fig. 5** The LR-ATC Algorithm pseudocode

using ATC. This is described in the ATC Subroutine pseudocode defined in Fig. 6. Once the ATC code terminates, the solution is recorded and the objective function value is recorded as the current lower bound,  $CLB$ , solution. Next, the relaxed constraint is checked with the recorded solution from the RMP to check if it was violated. If the constraint was not violated, that means that the solution of the RMP is feasible to the original formulation and the objective function value is recorded as a potential upper bound value,  $PotUB$ . If the constraint was violated, then a heuristic model is solved to obtain a feasible solution to the original problem. This is described in the Heuristic Subroutine presented in Fig. 7.

Once a feasible solution is obtained, the objective function value is calculated and it is recorded as the  $PotUB$ . If

**Algorithm 2:** ATC Subroutine

```

Data:  $T_v, \varepsilon_R, \varepsilon_y$ 
Result: Solution  $(R_v, \tilde{X}_V, y_s, R_s)$  to RMP
begin
  Solve Top-Level problem without deviation constraints;
  repeat
    for  $level : i \in 2..N, 1$  do
      Solve all subproblems in level  $i$ 
      |  $R_v^i \leftarrow \tilde{R}_v$ 
      |  $y_s^i \leftarrow \tilde{y}_s$ 
    end
  until  $(\varepsilon_R, \varepsilon_y < 0.005)$ ;
end

```

**Fig. 6** The ATC Subroutine

**Algorithm 3:** Heuristic Subroutine

```

Data:  $\tilde{x}_j, \forall j \in J$ 
Result: Solution  $(x_j)$  to Heuristic Model
begin
  Solve Heuristic1 fixing  $\tilde{x}_j$  as input;
  if Heuristic1 is feasible then
    |  $PotUB \leftarrow COFV(Heuristic1)$ ;
  else
    Solve Heuristic2 fixing  $\tilde{x}_k$  for  $K \subseteq J$ ; if Heuristic2 is feasible
    then
      |  $PotUB \leftarrow COFV(Heuristic2)$ ;
    else
      |  $PotUB \leftarrow \infty$ 
    end
  end
end

```

**Fig. 7** The Heuristic Subroutine

the value of the  $PotUB$  is lower than the  $UB$ , then the  $UB$  is updated with the objective function solution for the heuristic model. Otherwise, the  $UB$  value is not updated and the  $noimp$  parameter is incremented by 1. Next, the stepsize is calculated as in Eqn. 5, where  $\delta$  corresponds to the constraint violation. If the constraint is active, then the stepsize is set as 0. However, if the constraint is violated or inactive, then the stepsize is calculated based on the values of the  $UB$  and the  $CLB$  as well as the slack or surplus from the relaxed constraint. The scale factor also impacts the stepsize value and as the number of iterations increases, the stepsize is decreased by the stepsize value. This allows the model to fine tune the change in  $\lambda$  as the constraint activity is reached as shown in Eqn. 6. As seen in the constraint,  $\lambda$  can either increase or decrease depending on whether the relaxed constraint was violated or not ( $\delta$  can either be positive or negative). Lastly, the  $LB$  and the stepsize are updated based on whether there is an improvement in the lower and upper



bounds and if the *noimp* parameter is greater than 5. The algorithm terminates when the conditions of the while loop are no longer met.

$$\alpha = \frac{\epsilon(UB - CLB)}{\sqrt{\delta^2}} \tag{5}$$

$$\lambda_i = \max\{0, \lambda_{i-1} + \alpha\delta\} \tag{6}$$

The ATC Subroutine is detailed in Fig. 6, for a general problem decomposition of N levels and each level can have multiple subproblems. The notation in the algorithm is based on the work in Kim et al. (2003), where  $T_v$  denotes the design targets, and  $\epsilon_v$  and  $\epsilon_y$  denote the deviation tolerance for coordinating the system responses and the and system linking variables. The problem formulation is set up as in Kim et al. (2003) and the ATC algorithm is initialized by solving the top level optimization problem without including the deviation constraints (since no lower level response has been recorded yet). The algorithm then iterates between levels, where all subproblems in a single problem are solved simultaneously with deviation targets for the shared variables between them, if any. In each iteration, the values of the shared lower and upper levels are shared so as to ensure that the discrepancy between their values is minimized. This pseudocode focuses on the traditional ATC method, for the AL ATC and the AL-AD ATC, the readers are referred to the relevant references Tosserams et al. (2006), Tosserams et al. (2005).

As for the heuristic algorithm that is shown in Fig. 7, an example of a generalized heuristic solution scheme is presented. Although any heuristic method could be used, in order to present a solution scheme that can be easily applied to all applications, a simple procedural method is proposed. The method in Fig. 7, starts off by selecting a subset of decision variables whose values have been determined from the RMP, and setting them as parameters in the original formulation (before constraint relaxation). This is referred to as the first heuristic model. In terms of the decision variables that are fixed, although any combination can be selected, in order to leverage the decomposable nature of the problem, the shared variables between subproblems can be fixed. This allows the subproblems to be solved completely in parallel, which will drastically decrease the computational time required to obtain a feasible solution. Any resulting solution can be suboptimal to the original formulation, however, if it is feasible to the heuristic model, then it is feasible to the original formulation. If a feasible solution is obtained, the objective function value is stored as a *PotUB* for the problem and the LR-ATC scheme is continued. On the other hand, in the case that the problem has an empty feasible solution space, the second heuristic model can be solved. This is similar to the first heuristic model, however, a smaller subset of decision variables is fixed in order to

increase the feasible region to the problem. The advantage of this method is that as the LR outer loop approaches a feasible solution while refining the Lagrangian penalty parameter, the heuristic solution also improves with the improving values of the decision variables from the RMP. While any number of heuristic models can be introduced until a feasible solution is obtained, in order to limit the number of inner loop iterations, the heuristic models are limited to two models.

## 4 Numerical case studies

### 4.1 Geometric programming problem

In this section, a numerical example is used to illustrate the benefit of the proposed nested solution scheme in handling complex dependencies between subsystems. The proposed LR-AL ATC solution method is applied to the numerical example and is shown to perform well for even the simple numerical example presented in this section. The results show that the advantage of the solution scheme over the existing methods in the literature.

#### 4.1.1 Problem description

The numerical example used in this study is based on the non-convex geometric programming problem formulation presented in Kim et al. (2006). However, one of the contributions of the proposed method is its applicability to sub-problems that are highly coupled. Therefore, the a constraint in the model has been modified to further increase the coupling in the numerical model. Specifically, Eqns. 7 represent the updated formulation for the non-convex, geometric programming model presented in Kim et al. (2006), where Eqn. 7b has been updated from Eqn. 8.

$$\min f = x_1^2 + x_2^2 \tag{7a}$$

$$\text{s.t. } g_1 : \frac{x_3^{-2} + x_4^2}{x_5^2} \leq 1 \tag{7b}$$

$$g_2 : \frac{x_9^2 + x_6^{-2}}{x_7^2} \leq 1 \tag{7c}$$

$$g_3 : \frac{x_8^2 + x_9^2}{x_{11}^2} \leq 1 \tag{7d}$$

$$g_4 : \frac{x_8^{-2} + x_{10}^2}{x_{11}^2} \leq 1 \tag{7e}$$

$$g_5 : \frac{x_{11}^2 + x_{12}^{-2}}{x_{13}^2} \leq 1 \tag{7f}$$

$$g_6 : \frac{x_{11}^2 + x_{12}^2}{x_{14}^2} \leq 1 \tag{7g}$$

$$R_1 = x_1 = r_1(x_3, x_4, x_5) = (x_3^2 + x_4^{-2} + x_5^2)^{1/2} \tag{7h}$$

$$R_2 = x_2 = r_2(x_5, x_6, x_7) = (x_5^2 + x_6^2 + x_7^2)^{1/2} \tag{7i}$$

$$R_3 = x_3 = r_3(x_8, x_9, x_{10}, x_{11}) = (x_8^2 + x_9^{-2} + x_{10}^2 + x_{11}^2)^{1/2} \tag{7j}$$

$$R_4 = x_6 = r_4(x_{11}, x_{12}, x_{13}, x_{14}) = (x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2)^{1/2} \tag{7k}$$

$$x_i \geq 0, \quad \forall i \in 1 - 14$$

$$g_2 : \frac{x_5^2 + x_6^{-2}}{x_7^2} \leq 1 \tag{8}$$

This model is used since it provides a simple numerical example for a non-linear problem with non-trivial coupling between subsystems. Such problems are underrepresented in the literature and need tools that can handle the intricate relationships between the sub-models. For this reason, the numerical example presented in Eqns. 7 is used to illustrate the proposed method’s ability to handle problems that are more representative of real-world, complex systems.

### 4.1.2 Heuristic models

In order to generate quick, feasible solutions to the original problem, heuristic models are used. However, in order to bridge the gap between the lower-bound solution (the solution to the Relaxed Master Problem), there needs to be a link between the heuristic and the solution to the RMP. For this reason, three heuristic models are introduced that utilize partial results from the solution of the RMP. The reason that multiple heuristics are generated, is due to the fact that in some cases, utilizing the results of the RMP in the original problem could result in an infeasible solution depending on if the introduced constraints are within the feasible region for the original problem, which tends to be smaller than that of the RMP.

The first heuristic is divided into three models that can be solved in parallel. This parallel solution scheme is achieved by fixing the values of the linking variables between the different sub-models. The first heuristic model is shown in Eqns. 9, Eqns. 10 and Eqns. 11. The three models make up the heuristic where the objective value can be calculated

from just solving Eqns. 9, and the values of the remaining decision variables can be determined by solving all three sub-models. The linking variable values selected are  $x_3$ ,  $x_6$ , and  $x_{11}$ . Given that the values of these variables are fixed to be the result of the corresponding variables in the RMP, then the three sub-models can be solved individually on different machines or in parallel. This greatly reduces the computational cost of retrieving a feasible solution to the original problem. The objective functions of 10 and 11 are set to 1 in order to get the first feasible solution available for the variable values that are not in the objective of the original problem.

$$\min f = x_1^2 + x_2^2 \tag{9a}$$

$$\text{s.t. } g_1 : \frac{x_3^{-2} + x_4^2}{x_5^2} \leq 1 \tag{9b}$$

$$R_1 = x_1 = r_1(x_3, x_4, x_5) = (x_3^2 + x_4^{-2} + x_5^2)^{1/2} \tag{9c}$$

$$R_2 = x_2 = r_2(x_5, x_6, x_7) = (x_5^2 + x_6^2 + x_7^2)^{1/2} \tag{9d}$$

$$x_i \geq 0, \quad \forall i \in 1 - 7$$

$$\min f = 1 \tag{10a}$$

$$\text{s.t. } g_3 : \frac{x_8^2 + x_9^2}{x_{11}^2} \leq 1 \tag{10b}$$

$$g_4 : \frac{x_8^{-2} + x_{10}^2}{x_{11}^2} \leq 1 \tag{10c}$$

$$R_3 = x_3 = r_3(x_8, x_9, x_{10}, x_{11}) = (x_8^2 + x_9^{-2} + x_{10}^2 + x_{11}^2)^{1/2} \tag{10d}$$

$$x_i \geq 0, \quad \forall i \in 3, 8 - 11$$

$$\min f = 1 \tag{11a}$$

$$\text{s.t. } g_5 : \frac{x_{11}^2 + x_{12}^{-2}}{x_{13}^2} \leq 1 \tag{11b}$$

$$g_6 : \frac{x_{11}^2 + x_{12}^2}{x_{14}^2} \leq 1 \tag{11c}$$

$$R_4 = x_6 = r_4(x_{11}, x_{12}, x_{13}, x_{14}) = (x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2)^{1/2} \tag{11d}$$

$$x_i \geq 0, \quad \forall i \in 6, 11 - 14$$



In the case that fixing the values of  $x_3$ ,  $x_6$ , and  $x_{11}$  results in an empty feasible space, then the second heuristic model can be solved. The second heuristic model consists of two sub-models and is represented in Eqns. 12 and 13. Note that the main difference between the Heuristic 1 and 2 is that the second and third sub-models in Heuristic 1 have been combined in one model. This means that the number of variables to be fixed can be reduced to only fixing  $x_3$  and  $x_6$ . This increases the search space within the feasible region and improves the probability of finding a feasible solution.

$$\min f = x_1^2 + x_2^2 \tag{12a}$$

$$\text{s.t. } g_1 : \frac{x_3^{-2} + x_4^2}{x_5^2} \leq 1 \tag{12b}$$

$$R_1 = x_1 = r_1(x_3, x_4, x_5) = (x_3^2 + x_4^{-2} + x_5^2)^{1/2} \tag{12c}$$

$$R_2 = x_2 = r_2(x_5, x_6, x_7) = (x_5^2 + x_6^2 + x_7^2)^{1/2} \tag{12d}$$

$$x_i \geq 0, \quad \forall i \in 1 - 7$$

$$\min f = 1 \tag{13a}$$

$$\text{s.t. } g_3 : \frac{x_8^2 + x_9^2}{x_{11}^2} \leq 1 \tag{13b}$$

$$g_4 : \frac{x_8^{-2} + x_{10}^2}{x_{11}^2} \leq 1 \tag{13c}$$

$$g_5 : \frac{x_{11}^2 + x_{12}^{-2}}{x_{13}^2} \leq 1 \tag{13d}$$

$$g_6 : \frac{x_{11}^2 + x_{12}^2}{x_{14}^2} \leq 1 \tag{13e}$$

$$R_3 = x_3 = r_3(x_8, x_9, x_{10}, x_{11}) = (x_8^2 + x_9^{-2} + x_{10}^{-2} + x_{11}^2)^{1/2} \tag{13f}$$

$$R_4 = x_6 = r_4(x_{11}, x_{12}, x_{13}, x_{14}) = (x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2)^{1/2} \tag{13g}$$

$$x_i \geq 0, \quad \forall i \in 3, 6, 8 - 14$$

Given that both the first and the second heuristic models have an empty feasible region, the third heuristic model is solved. The disadvantage of the third model is that it is costly, although still less computationally expensive than the original problem. The third heuristic model is described in Eqns. 14. Here, only one model is solved, since the equations are linked

as only  $x_3$ 's value is known. Therefore, none of the constraints can be decoupled.

$$\min f = x_1^2 + x_2^2 \tag{14a}$$

$$\text{s.t. } g_1 : \frac{x_3^{-2} + x_4^2}{x_5^2} \leq 1 \tag{14b}$$

$$g_3 : \frac{x_8^2 + x_9^2}{x_{11}^2} \leq 1 \tag{14c}$$

$$g_4 : \frac{x_8^{-2} + x_{10}^2}{x_{11}^2} \leq 1 \tag{14d}$$

$$g_5 : \frac{x_{11}^2 + x_{12}^{-2}}{x_{13}^2} \leq 1 \tag{14e}$$

$$g_6 : \frac{x_{11}^2 + x_{12}^2}{x_{14}^2} \leq 1 \tag{14f}$$

$$R_1 = x_1 = r_1(x_3, x_4, x_5) = (x_3^2 + x_4^{-2} + x_5^2)^{1/2} \tag{14g}$$

$$R_2 = x_2 = r_2(x_5, x_6, x_7) = (x_5^2 + x_6^2 + x_7^2)^{1/2} \tag{14h}$$

$$R_3 = x_3 = r_3(x_8, x_9, x_{10}, x_{11}) = (x_8^2 + x_9^{-2} + x_{10}^{-2} + x_{11}^2)^{1/2} \tag{14i}$$

$$R_4 = x_6 = r_4(x_{11}, x_{12}, x_{13}, x_{14}) = (x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2)^{1/2} \tag{14j}$$

$$x_i \geq 0, \quad \forall i \in 1 - 14$$

Note that the three heuristic models are arranged based on the computational cost needed to solve them, starting with the heuristic that has the smallest feasible region, to the heuristic model with the largest feasible region (if any exists). This is done intentionally to reduce the computational cost required to retrieve an updated upper bound for the minimization problem. In the case that all three heuristic models have an empty feasible region, the previous upper bound (for a minimization problem) is used.

### 4.1.3 Results

The numerical model was solved using ATC, AL ATC, AL-AD ATC, LR and LR-AL ATC. A comparison of the results is provided in Table 1, where the objective value of the problem, the computational time, as well as the number of iterations, are compared. The purpose of this case study is

to validate the proposed method using a common numerical case study from the MDO literature. The number of iterations is compared for the inner loop as well as the outer loop and the total number of iterations is detailed in the table. The inner and outer loop definition changes based on the solution algorithm used. In ATC, the iterative scheme aims to coordinate the shared values across the decomposed subproblems, therefore, it is a single-loop method where the targets for variable discrepancies are updated. In the AL ATC method, the inner loop coordinates between the different subproblems as in the ATC method. The main difference lies in the formulation of the penalty term, so the total number of iterations is equivalent to the number of iterations in the inner loop. For the AL-AD ATC method, since the subproblems are solved once in each iteration, the number of iterations in the outer loop are determined as the number of times all three subproblems are solved. As for the LR method, it is a single-loop solution scheme (assuming that the UB heuristic is not an iterative method) where the value of the penalty parameter is updated in each iteration and the new upper bound and lower bound values are calculated until the termination criteria is reached. The LR-AL ATC method on the other hand is a double-loop method where the outer loop is the LR loop to update the value of the penalty parameter, and the inner loop is an AL ATC loop that aims to reduce discrepancies between the decomposed subproblems. The computational cost is calculated as the total solution time in GAMS modeling software using the *Time Elapsed* feature.

To compare the results of the proposed algorithm against existing methods, the objective function values are compared. In this case, the proposed LR-ALAD ATC solution method is able to achieve the same result as the AL ATC method, however, only the LR algorithm is able to achieve an objective value equivalent to the AAO solution. The ATC method attains the worst objective value since it indicates an infeasible solution (has an objective value that is lower than the optimal value of a minimization problem). The reason for the infeasible solution is that the model was terminated at 300 iterations due to limited computational power. Besides the ATC method, all other methods return close to feasible

solutions, with AL-ATC and ALAD ATC having less optimal results. This is due to the difficulty when coordinating between the subproblems when strong coupling exists between them.

Although the number of iterations is compared in Table 1, the computational cost per iteration changes drastically from one algorithm to the next. Therefore, the total computational time is compared alongside the number of inner and outer loop iterations. The computational time recorded is obtained from the *Time Elapsed* feature in GAMS. In this case, the computational time is compared for the overall solution scheme until convergence or termination of the algorithm. The best solution is achieved from the LR model. Not only is it able to obtain an optimal solution, but it was also able to obtain it within a single iteration and it took a total of around 1 s. This is much faster than even the AAO solution, even for a simple example. Although ATC was able to achieve the best results when compared with the rest of the methods, it is expected for such a simple example where the entire problem can be solved to optimality within 4 s. ATC methods show reduced efficiency when compared with relaxation methods and an AAO approach. This is also expected for such a simple problem when introducing an iterative approach, since the benefit of decomposition does not outweigh the cost introduced by iterating through the method. Moreover, this is justified by the fact that a complicating constraint, Eqn 7c, exists, where it increases the coupling between the subproblems and increases the number of iterations needed to reach a feasible solution to the original problem. A clear improvement is seen when integrating the relaxation method with a decomposition method such as ATC. Although in some cases the number of iterations could be higher for the integrated approach, such as comparing the total iterations for the AL ATC method and that of the LR-ATC method, the computational cost for the integrated approach is greatly reduced from over 14,000 s to approximately 241 s. This is attributed to the fact that a single iteration of the integrated approach is expected to be faster since three simple problems are solved as opposed to a single difficult problem. This improvement is even evident on a simple problem

**Table 1** Comparison of Results For Different Solution Schemes

Model	Objective Value	Iteration Count			Computational Time (s)
		Inner Loop	Outer Loop	Total	
AAO	17.454	–	–	–	3.063
ATC	8.232	–	300*	300	327.781
AL ATC	18.611	300*	1	300	14687.75
AL-AD ATC	18.009	–	300*	300	416.455
LR	17.455	–	1	1	1.071
LR-ATC	17.455	300*	2	601	240.755
LR-AL ATC	17.5	9	2	11	10.689
LR-ALAD ATC	17.608	–	18	18	54.918

such as the numerical example studied here. The benefits are expected to be greater for more complicated problem structures.

In Table 2, the optimal solution is compared for the different approaches. Most of the methods provide solutions close to the optimal solution of the original problem, with the LR and LR-ATC methods providing optimal solutions. However, the ATC, AL-ATC and ALAD ATC methods have a reduced efficiency when compared with the other methods. This is mostly due to the added coupling between the decomposed subproblems that make coordinating between the problems difficult and add more constraints to achieving a feasible solution to the AAO formulation. If we are to take closer look at the decision variable results, we could see that for the methods with objective function values further from the optimal solution, the variables that have the biggest discrepancies from their optimal values are mostly the shared variables along the decomposed problems,  $x_3, x_6, x_9, x_{11}$ . This pattern is evident in the results of the ATC method, where  $x_{11}$  has the biggest deviation from the optimal value and it is shared among the three decomposed problems and the model had difficulty coordinating between them. This reinforces the benefit of the proposed method, since it can relax complicating, coupling constraints to ensure that the embedded decomposition scheme can obtain close to optimal solutions for the relaxed problem. Then, the relaxed constraints are optimized in the outer loop. Therefore, we can see improved solutions for the relaxation schemes that are integrated with modified versions of the ATC method.

### 4.2 Scalability study

The following subsection presents a numerical example to study how well the proposed solution scheme scales with the uncertainty. The problem formulation is presented in Sect. 4.2.1 and the results of the comparison study are presented in Sect. 4.2.2.

#### 4.2.1 Problem formulation

The proposed numerical problem is presented in Eqns. 15a to 15i. This numerical formulation has two characteristic problem groupings. The proposed decomposition scheme is presented in Fig. 8 and the two subproblems are highlighted. The figure shows the benefit of component aggregation through relaxation of complicating constraints and how the relaxation can provide a benefit over traditional MDO techniques.

$$\min f = x_1^2 + x_1x_2 + x_4x_5 + x_3^2 \tag{15a}$$

$$\text{s.t. } -\frac{x_1^2}{5} + \frac{4x_1x_2}{3} - \frac{6x_2^2x_1}{5} - 6 \leq 0 \tag{15b}$$

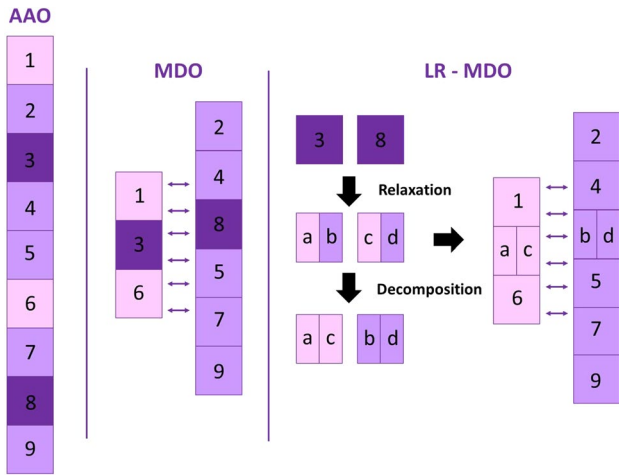
$$-\frac{4x_4x_5}{3} - \frac{2x_3^2}{3} - \frac{3x_3x_4}{8} - 5 \leq 0 \tag{15c}$$

$$\frac{x_1}{3} + \frac{x_2}{2} - \frac{4x_3}{3} \leq 0 \tag{15d}$$

**Table 2** Comparison of results for the modified numerical example

	AAO	ATC	AL-ATC	ALAD ATC	LR	LR-ATC	LR-AL ATC	LR-ALAD ATC
UB	–	–	–	–	17.455	17.455	17.5	17.608
LB	–	–	–	–	17.455	17.455	11.602	17.455
$z$	17.4548	8.232	18.611	18.009	17.455	17.455	17.5	17.608
$X_1$	2.8529	2.183	3.472	3.36	2.853	2.852	2.802	2.852
$X_2$	3.0522	1.863	2.561	2.593	3.052	3.052	3.106	3.053
$X_3$	2.418	2.336	3.151	3.151	2.418	2.417	2.356	2.416
$X_4$	0.8409	0.841	0.841	0.841	0.841	0.841	0.841	0.841
$X_5$	0.9371	1.079	0.899	0.904	0.937	0.937	0.942	0.937
$X_6$	2.7731	1.793	2.299	2.326	2.773	2.774	2.815	2.774
$X_7$	0.8647	0.94	0.681	0.704	0.865	0.865	0.915	0.866
$X_8$	1.0014	1.014	0.995	0.994	1.001	1.001	0.993	1.001
$X_9$	0.7859	0.848	0.52	0.523	0.786	0.787	0.843	0.787
$X_{10}$	0.7895	0.88	0.501	0.532	0.790	0.790	0.827	0.79
$X_{11}$	1.273	0.358	1.123	1.318	1.273	1.273	1.303	1.274
$X_{12}$	0.8409	0.481	0.842	0.841	0.841	0.841	0.841	0.841
$X_{13}$	1.742	1.242	1.495	1.526	1.742	1.742	1.764	1.743
$X_{14}$	1.5257	0.914	1.237	1.246	1.526	1.526	1.551	1.52

\* The maximum number of iterations was set at 300



**Fig. 8** Decomposition for the numerical example with the numbered blocks corresponding to the constraints in Eqns. 15a to 15j

$$-\frac{x_5}{2} - \frac{5x_3x_5}{3} - x_4^2 - 3 \leq 0 \tag{15e}$$

$$x_5^2 + x_4 - x_3x_4 - 2 \leq 0 \tag{15f}$$

$$x_3 - 10x_4 \geq 2r_t^s \tag{15g}$$

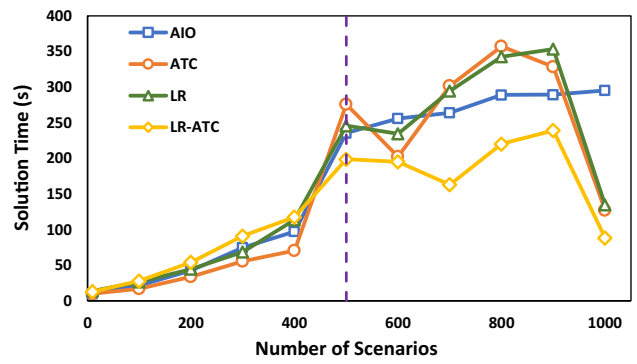
$$x_4 = x_5 + 0.6 \tag{15h}$$

$$x_1 = x_4 + x_5 - x_3^2 + 3 \tag{15i}$$

$$x_1 \geq 10q_t^s \tag{15j}$$

**4.2.2 Results**

In order to test the impact of increasing the size of the problem on the efficiency of the proposed algorithm. The mathematical example is used to vary the number of parameters in the model as well as decision variables and test how well it scales computationally. It is solved with the proposed solution scheme and compared with an ATC decomposition scheme, an LR solution scheme, and the AAO approach. The same set of randomly generated scenarios are used for all the problems in order to ensure a fair comparison. Additionally the same constraints are relaxed for both LR and the LR-ATC method. In terms of the decomposition, the scheme presented in Sect. 4.2 is used for both the ATC method and the proposed LR-ATC method.



**Fig. 9** Scalability of the proposed solution method with the size of the problem

Figure 9 shows the results comparing the proposed solution method with the All-In-One (AIO) approach, using ATC for decomposition and LR for relaxation, separately. The results show that as the number of scenarios considered for the stochastic parameters increases, so does the computational benefit of the proposed approach. After 500 scenarios, the proposed hybrid approach outperforms the other solution methods. The benefit of the proposed scheme becomes more evident against the other methods. This is due to the fact that the cost of coordinating the subproblems is no longer dominating when compared with the benefit of decomposing the problem when the size of the problem increases. This is for the case where no parallelization is used to solve the subproblems. In the case that parallelization is to be used, the benefit of the proposed scheme increases drastically.

**5 Golinski speed reducer problem**

**5.1 Problem description**

The Golinski Speed Reducer problem is a popular problem from the NASA Langley center for MDO problems Ray (2003) for which the schematic is shown in Fig. 10. The model aims to minimize the volume of the speed reducer, while accounting for physical system constraints Tosserams et al. (2007). The main design variables considered include designing the shaft and gear diameters.

**5.2 Problem formulation**

The formulation of the problem is extracted from Tosserams et al. (2007), for the multidisciplinary problem. In order to solve the MDO problem using a hierarchical solution scheme such as ATC, the subproblems have to be linked hierarchically. The proposed hierarchical decomposition scheme is presented in Fig 11. Here the model consists of three levels and each level has only one single subproblem.

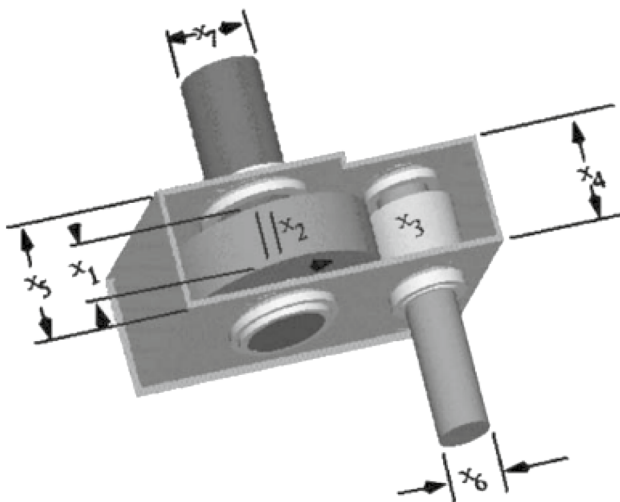


Fig. 10 Schematic of the Golinski Speed Reducer from Tosserams et al. (2007)

The decomposition into three subproblems is based on the scheme presented in Tosserams et al. (2007) but modified in order to be solved hierarchically.

### 5.3 Results

The Golinski Speed Reducer problem was solved using an AAO approach and compared with an ATC solution scheme, an AL-ATC solution scheme and an ALAD solution scheme. In all cases, the hierarchical structure depicted in Fig 11 is utilized. The results are presented in Tables 3 and 4, where the objective value, decision variable values and number of iterations are compared for each of the methods studied. The different models were coded using GAMS modeling software and BARON solver was used for Non-linear Programming models. In order to limit the computational time to a reasonable time, the maximum number of iterations was set to be 300 iterations and the best feasible solution identified up to that iteration was used.

Table 3 shows a comparison of ATC methods with the AAO formulation for the Golinski Speed Reducer problem. As expected, the ALAD ATC method is able to reach a solution within only 85 iterations, which is greatly reduced as compared with the number of iterations of the ATC and AL-ATC methods. The ATC method needed more than 300 iterations to converge, however, the results closely matched the AAO results for the Golinski Speed Reducer problem. The reduced computation of the AL-ATC and ALAD ATC methods have a much reduced computational time and are still able to provide results that are close to optimal. Not only is the difference in objective values over 3.2%, the decision variable results also varied drastically for  $X_6$ , where the percentage difference was

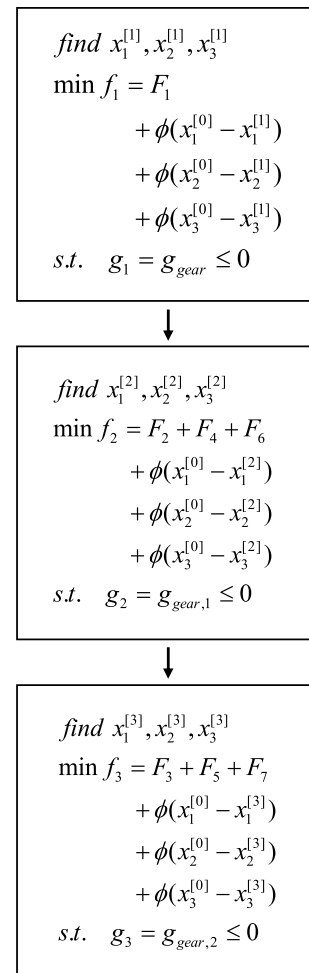


Fig. 11 Hierarchical decomposition scheme for the Golinski Speed reducer problem

Table 3 Comparison of Results for the Golinski Speed Reducer Problem using ATC

	AAO	ATC	AL-ATC	ALAD ATC
Objective Value	2894.12	2987.755	2988.361	2988.361
$X_1$	3.5	3.6	3.6	3.6
$X_2$	0.7	0.8	0.7	0.7
$X_3$	17	17.111	17	17
$X_4$	7.3	7.3	7.3	7.3
$X_5$	7.715	7.715	7.715	7.715
$X_6$	2.9	3.349	3.350	3.350
$X_7$	5.287	5.286	5.286	5.287
Iteration Count	–	300*	19	18
Computation Time (s)	4.000	393.563	16.109	17.312

\* The maximum number of iterations was set at 300

**Table 4** Comparison of Results for the Golinski Speed Reducer Problem with LR approaches

	AAO	LR	LR-ATC	LR-ALAD ATC
UB	–	2994.267	3538.889	2897.943
LB	–	2894.187	2888.522	2897.943
Objective Value	2894.187	2894.187	3538.889	2897.943
$X_1$	3.5	3.5	3.6	3.255
$X_2$	0.7	0.7	0.8	0.700
$X_3$	17	17	17.034	17
$X_4$	7.3	7.3	7.3	7.3
$X_5$	7.715	7.715	7.715	7.715
$X_6$	2.9	2.9	3.349	3.350
$X_7$	5.287	5.287	5.286	5.287
Iteration Count	–	96	192	6
Computational Time (s)	4.000	113.375	108.282	10.156

over 20% for all the ATC methods. Although ATC did not converge within 300 iterations, its solution more closely matched the AAO solution to the problem. Although it should be noted that all the results were within 4%.

As for the proposed solution scheme, it was compared with the AAO solution for the Golinski Speed Reducer problem as well as LR and LR-ATC in Table 4. For the LR methods, the same constraint set was relaxed to allow for fair comparison of the models. The relaxed constraints correspond to constraints  $g_1$  and  $g_8$  in the formulation of the Golinski Speed Reducer problem in Tosserams et al. (2007). These constraints are shown in Eqns. 16 and 17, since the LR algorithm depicted in Fig 5 provides a lower and upper bound, when the algorithm terminates, if the lower bound is feasible to the original problem (for a minimization problem), then it is taken as the objective value. However, if the lower bound is not feasible, then the upper bound (which should always be feasible to the original formulation) is taken as the objective value. The results of the comparison of LR methods show that LR alone needs less iterations than ATC methods, especially since only two constraints are relaxed. Combining LR with ATC, we see an improvement in the iteration count for ATC, however, in this case, the objective value was much lower than the objective value for the AAO solution, with some discrepancies in the optimal values of the decision variables. In this case, the use of ATC within the LR scheme did not perform better than using LR alone in terms of the iteration count and the objective value alone. In fact, the difference in the objective value was over 22%.

The results were also compared for the computational time for this simple example. It should be noted however, that for a small-scale example, we expect the presented

iterative methods to take less time than the AAO solution. However, for a simple solution such as this, we expect that the AAO will have the least computation time. We compare the computational time of the algorithms for this small example and expect to see an exaggerated trend for larger problems. It should also be noted that although the LR-ATC method actually needed more iterations to terminate than the LR approach, it took less time than LR since a single iteration took less time when solving the decomposed ATC problem. In fact, the combined LR-ATC method improved the computational time by approximately 5%, at the cost of a 23% deterioration in the quality of the objective function value. The reduced quality of the objective function value is overcome by utilizing the proposed LR-ALAD ATC solution approach. In this case, the objective function is within 0.2% of the optimal solution obtained from the AAO approach. Moreover, the computational time is further improved to over 91% of the LR method alone.

$$g_1 : \frac{1}{110x_6^3} \sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 1.69 * 10^7} - 1 \quad (16)$$

$$g_8 : \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \quad (17)$$

In terms of the proposed LR-ALAD ATC method, the number of iterations is greatly reduced to only 6 iterations as compared with 96 iterations by using LR alone. Moreover, although the objective function value is not comparable to the LR method alone, since the ALAD ATC method embedded in LR allows for paralleling the subproblems and decomposing them into smaller problems, one run of LR should be faster for the proposed scheme, as evident by the computational time. In fact, in the worst case, a single iteration of the LR-ALAD ATC method should be drastically shorter than a single iteration of the LR method, or the problem decomposition is not optimal. Overall, the proposed method can be seen to greatly reduce the number of iterations for the LR method while maintaining a close accuracy of the results.

In this case, when comparing the LR-ALAD ATC method with the ALAD ATC method, not only is the objective function value improved slightly (approximately 3%), the number of iterations is also diminished by over 66.6% and the computational time decreased by 41.3%. When comparing with AL-ATC, the results follow a similar trend, with an equivalent improvement in the objective function value for the proposed solution scheme and a time improvement of approximately 37%. These results are expected to be even more improved for problems with complex structures where decomposing the problem should result in drastic improvements on the computational time.



## 6 Conclusions

A novel hybrid solution scheme that decomposes complex constraints across subproblems in MDO problems is presented for coupled non-decomposable systems. The proposed method integrates the benefits of utilizing an outer relaxation solution scheme such as Lagrangian Relaxation while leveraging MDO methods such as Analytical Target Cascading and its variations to decompose the problem. A generalized framework is presented for such problems and the proposed methodology has been applied to a numerical problem as well as the Golinski Speed Reducer problem set provided by NASA Langley. The solution scheme was compared with several variations and hybrid algorithms based on LR and ATC and the proposed scheme is shown to drastically reduce the computational cost of the solution scheme while maintaining close to optimal solutions. An advantage of utilizing LR-based schemes is that an optimality gap based on the relaxed formulation can be used to provide an estimate of how far the feasible solution is from the best possible estimate. Moreover, a major advantage of ATC is that the complex system can be decomposed, and based on the application and target problem, in some cases that could allow for parallelisation when solving the subproblems. ALAD ATC also allows odd levels and even levels to be parallelised independently. For large applications the benefit is expected to be further heightened.

The proposed methodology has some limitations when implementing it for real-case studies. One limitation is that it is not always straightforward to identify the decomposability of mathematical models with strong coupling. Future work includes identifying systematic methods to identify different decomposition hierarchies for problems. Another limitation of the proposed work is the selection of the MDO method used in the proposed integrated scheme. As seen in the results, some integration schemes could be less optimal than others. There is a need to identify the best MDO solution scheme for different problems to ensure that the integrated solution approach is optimal.

For future work, the application of the proposed methodology to a time-staged application should be studied to quantify the benefit of the proposed methodology to time-dependent problems. The decomposition of the problem can be made based on the different time-dependent stages, where the connection between the different stages can be relaxed in the outer loop of the problem. This is contingent on an explicit relationship between stages, in the case of non-explicit relationships, surrogate modeling could be utilized to extract that relationship explicitly. Moreover, the proposed methodology should be applied to large-scale applications where traditional solution schemes are unable to handle the size and complexity of the problem structure.

Furthermore, it would be interesting to study and introduce a metric to measure the benefit of applying the proposed methodology to different problem structures. Although general guidelines can be set to determine how suitable the LR-ALAD ATC scheme is, a quantitative metric could be useful to estimate the benefit of using the method with only preliminary results.

**Acknowledgements** This research is partially supported by the National Science Foundation (NSF) through the Engineering Research Center for Power Optimization of Electro-Thermal Systems (POETS) with cooperative agreement EEC-1449548, and the Alfred P. Sloan Foundation through the Energy and Environmental Sensors program with Grant # G-2020-12455.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Replication of results** All the formulations presented in this paper are provided in the paper and can be replicated based on the available information in the paper. For the Golinski Speed Reducer problem, the data is provided in the references to allow replication of the results. GAMS modeling software was used to model the proposed integrated solution scheme and the solution scheme is provided in Section 3. The code is available upon request by contacting the corresponding author.

## References

- Afonso MV, Bioucas-Dias JM, Figueiredo MA (2010) An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems. *IEEE Trans Image Process* 20(3):681–695
- Fisher ML (1985) An applications oriented guide to Lagrangian relaxation. *Interfaces* 15(2):10–21
- Hamdan B, Diabat A (2020) Robust design of blood supply chains under risk of disruptions using Lagrangian relaxation. *Transp Res Part E* 134:101764
- Hamdan B, Ho K, Wang P (2019) Staged-deployment optimization for expansion planning of large scale complex systems. In: *AIAA Scitech Forum, 2019, American Institute of Aeronautics and Astronautics Inc, AIAA* (2019)
- Kang N, Kokkolaras M, Papalambros PY, Yoo S, Na W, Park J, Featherman D (2014) Optimal design of commercial vehicle systems using analytical target cascading. *Struct Multidisc Optim* 50(6):1103–1114
- Kim HM, Chen W, Wiecek MM (2006) Lagrangian coordination for enhancing the convergence of analytical target cascading. *AIAA J* 44(10):2197–2207
- Kim HM, Michelena NF, Papalambros PY, Jiang T (2003) Target cascading in optimal system design. *J Mech Des* 125(3):474–480
- Kim HM, Rideout DG, Papalambros PY, Stein JL (2003) Analytical target cascading in automotive vehicle design. *J Mech Des* 125(3):481–489
- Lizier JT, Bertschinger N, Jost J, Wibral M (2018) Information decomposition of target effects from multi-source interactions: perspectives on previous, current and future work. *Entropy* 20(4):307

- Martins JR, Kennedy GJ (2021) Enabling large-scale multidisciplinary design optimization through adjoint sensitivity analysis. *Struct Multidisc Optim* 64(5):2959–2974
- Papalambros PY, Wilde DJ (2000) *Principles of optimal design: modeling and computation*. Cambridge University Press, Cambridge
- Ramu P, Thananjayan P, Acar E, Bayrak G, Park JW, Lee I (2022) A survey of machine learning techniques in structural and multidisciplinary optimization. *Struct Multidisc Optim* 65(9):1–31
- Ray T (2003) Golinski's speed reducer problem revisited. *AIAA J* 41(3):556–558
- Santiago A, Huacuja HJF, Dorronsoro B, Pecero JE, Santillan CG, Barbosa JJG, Monterrubio JCS (2014) A survey of decomposition methods for multi-objective optimization. In: *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, pp 453–465. Springer, New York
- Tedford NP, Martins JR (2010) Benchmarking multidisciplinary design optimization algorithms. *Optim Eng* 11(1):159–183
- Tosserams S, Etman L, Papalambros P, Rooda J (2005) Augmented Lagrangian relaxation for analytical target cascading. In: *6th World Congress on Structural and Multidisciplinary Optimization*, May 30–June 3, Citeseer
- Tosserams S, Etman L, Papalambros P, Rooda J (2006) An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers. *Struct Multidisc Optim* 31(3):176–189
- Tosserams S, Etman LP, Rooda J (2007) An augmented Lagrangian decomposition method for quasi-separable problems in mdo. *Struct Multidisc Optim* 34(3):211–227
- Tosserams S, Kokkolaras M, Etman L, Rooda J (2010) A nonhierarchical formulation of analytical target cascading. *J Mech Des* 132(5):051002

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.