Presented in the 4th International Conference on Quantitative and Qualitative Methodologies in the Economic and Administrative Sciences, Editor: C.C. Frangos, Athens, Greece, 21-22 May 2015

Evolutionary Optimization of Simulation Workflows for Product Design Processes

Vassilios C. Moussas

Associate Professor, Dept. of Civil & Surveying Engineering, TEI of Athens, Greece

vmouss@teiath.gr

Abstract

Simulation workflow optimization has become an important investigation area, as it allows users to process large scale & heterogeneous problems in distributed environments in a more flexible way. The most characteristic categories of such problems come from the aerospace and the automotive industries. In this work a specially developed algorithm that is based on heuristic optimization techniques (Genetic Algorithms) is applied to deliver an optimized workflow implementation of an initial workflow schedule (PERT). In order to demonstrate its potentials, the algorithm is applied on a sample manufacturing product design problem that requires a lot of time consuming simulations & finite elements analysis under a constrained availability of computer resources.

Keywords: Simulation Workflow Optimization, PERT, Genetic Algorithms, Resource Optimization

1. Introduction

Workflows have been used to model repeatable tasks or operations in a number of different industries including manufacturing and software. A workflow typically represents a schedule of the required tasks (human, physical, virtual, etc.) based on their dependencies and the associated resources. Definitions of the term “workflow” can be found in Wikipedia and the [1]. Higher level workflows contain all tasks required to accomplish a specific goal. Some of these tasks are further analyzed in lower level workflows. Computation tasks are occupying an increasing part of the Product Design Process in industry today, often using supercomputers to calculate FEM or CFD analysis, or simulate large, physically complex systems modeled by PDEs. As a result the overall (higher) workflow efficiency and optimization is highly dependent by the optimization of the included (lower) simulation workflows.

An example workflow chart is presented below, redesigned from article [2], displaying the sequence of concatenated steps for weather prediction. In this higher level workflow, steps 1 to 3 represent the data collection from various sources and the necessary pre processing actions, steps 4 to 8 is the part containing the computational analysis and simulations (simulations workflow), and, steps 9 & 10 contain the post processing actions, statistical analysis and visualization of results. The simulations workflow contains several tasks or smaller workflows that perform a number of lengthy and complicated calculations such as: data transformation to a common format (in 4), numerical weather forecast based on mathematical-physical models, environmental area discretization, numerical solution of a system of partial differential equations (in 5), statistical interpretation and forecasting (in 6), numerical results interpolation (in 7) and, statistical post-processing to remove failures of measuring devices (e.g. using KALMAN filters) (in 8).

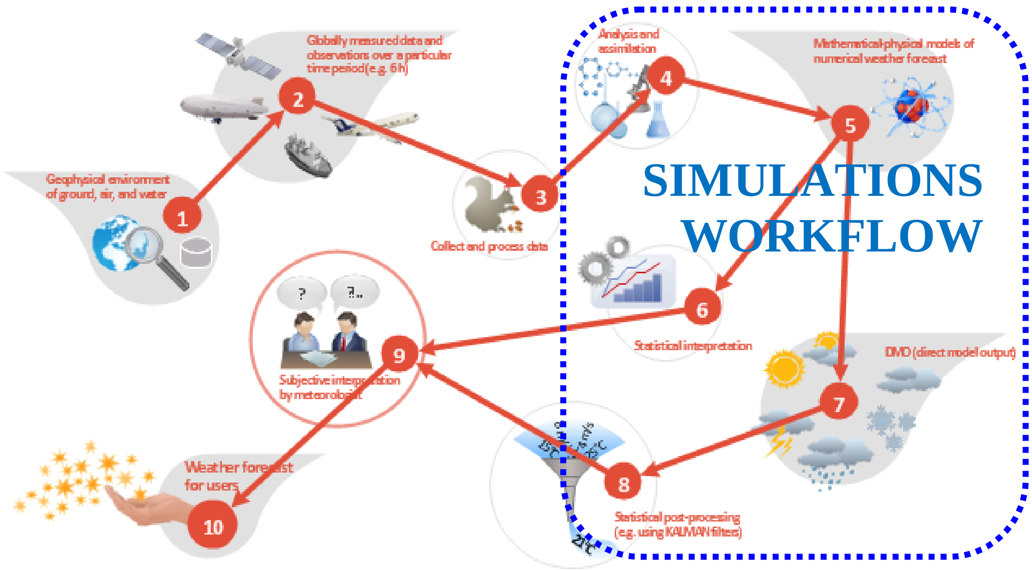


Figure 1. Example of a weather forecast workflow [2] and its simulation workflow part.

In the aerospace manufacturing industry we can find much more complex simulation workflows, as the one given in [3], where the manufacturer needs to solve a multi-disciplinary optimization problem involving several partners or subcontractors. In order to improve the design of a new cabin, a number of simulations & analysis tasks are required involving, calculations of environmental CFD models, structural FEM models, electrical/thermal control models, power plant FEM models and human response ANN models. The designer needs not only to optimize the final cabin design, but also to optimize the execution schedules of all the required computational tasks, in terms of time, cost and accuracy.

As the Product Development Processes (PDPs) become more and more decentralized and distributed, the associated optimization problems also become more complex, with multiple & contradictory objectives and they require powerful and/or specially designed optimization tools [4]. In addition, the distributed environment of large scale problems requires the software tools to be accessible from anywhere as been local. A promising simulation workflow optimization tool is proposed in [5] that is using evolutionary methods in order to optimize a heterogeneous simulation workflow containing several computational tasks that involve completely different software tools, resources, requirements and often contradictory objectives. Again the application comes from the aerospace manufacturing industry and the design of an aircraft tail rudder.

A typical experimental scenario is a repetitive cycle of moving data to a supercomputer for analysis or simulation, launching the computations and managing the storage of the output results. The scenario is often repeated hundreds of times in order to find an optimal solution. Scientific workflow systems aim at automating this cycle in a way to make it easier for scientists to focus on their research and not computation management. Several tools have been developed to support simulation workflows and automate the computational work involved. For instance, Optimus® by Noesis [6] is a complete tool that permits parametric modeling and also offers optimization techniques within its tool to enable simulation process integration, and to optimize engineering design and prototyping processes. As shown in figure 2a, an automobile manufacturing case may contain a simulation workflow that is introduced to the tool in order to be executed repeatedly until an optimal solution is found (figure 2b). The role of the optimization supported by these tools is to find the best product characteristics or the best design parameters to satisfy its requirements. The sequence of the computational tasks and the simulation execution details or synchronization are in general fixed from the initial workflow setup.

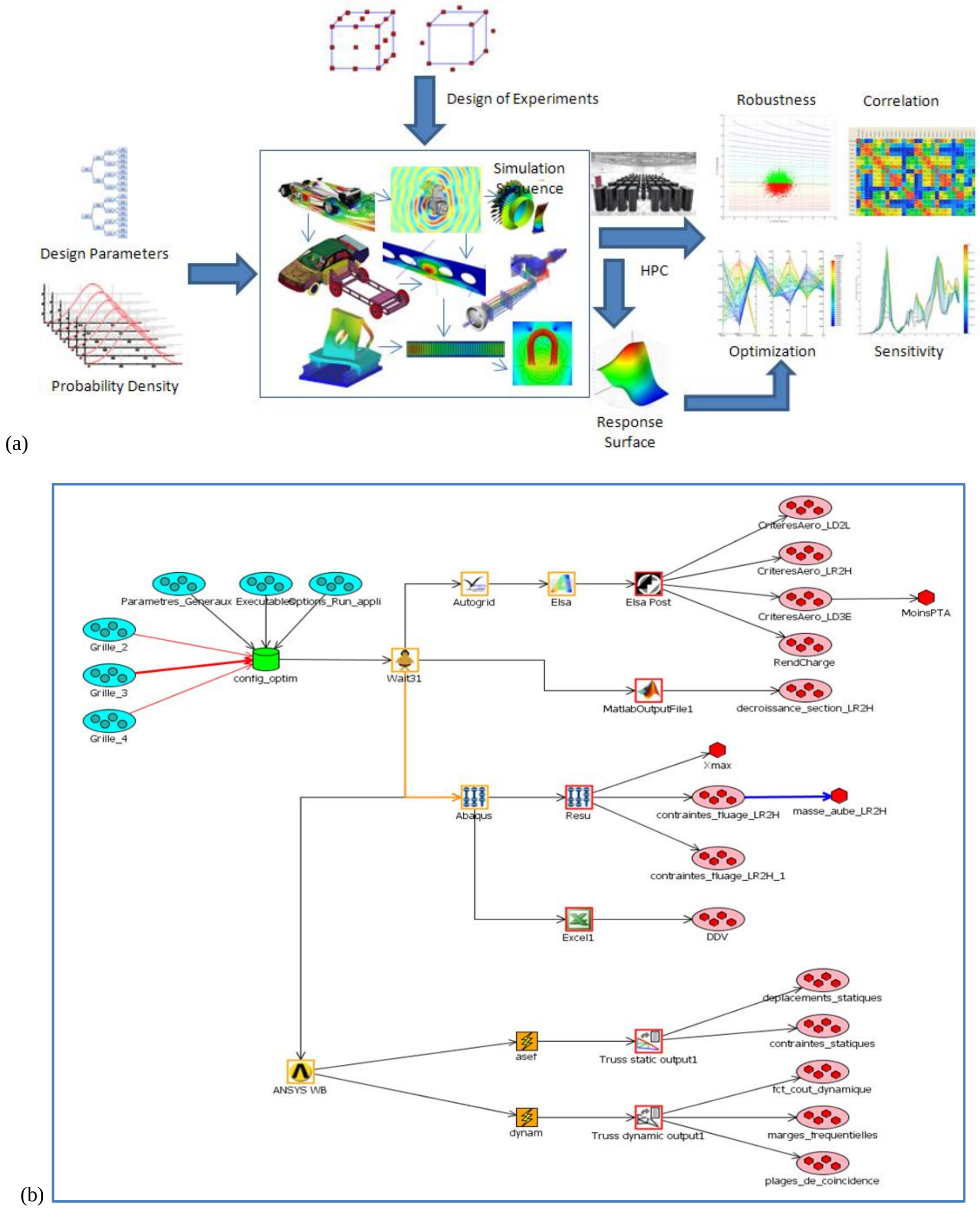


Figure 2. (a) A workflow from Automotive manufecturing, and, (b) the Simulation Workflow representation on Optimus Tool [6] for simulation automation & parameter optimization.

The required simulations may be executed in-house (in the same workstation, a dedicated server), or, remotely (a super computer, a cloud service, a sever farm, or another HPC web service). In the above example all runs are submitted to one resource (HPC) but the more realistic case is the availability of a set of resources with different performance, cost & utilization characteristics. The optimal scheduling of the simulation runs (on the available resources in the available time) is a completely different optimization task than the one supported by the workflow automation tools for the optimization of the product.

2. Problem Description

In the present work we will focus on the optimization of the workflow execution schedule and especially on a Resource-Constrained Optimization of a Simulation Workflow execution. More precisely, the aim is to design an optimization algorithm that provides optimal schedules for the execution of the computational tasks of a given workflow, satisfying not only the precision requirements but also time, money and resource constrains, as faced by a CAE designer (figure 3).

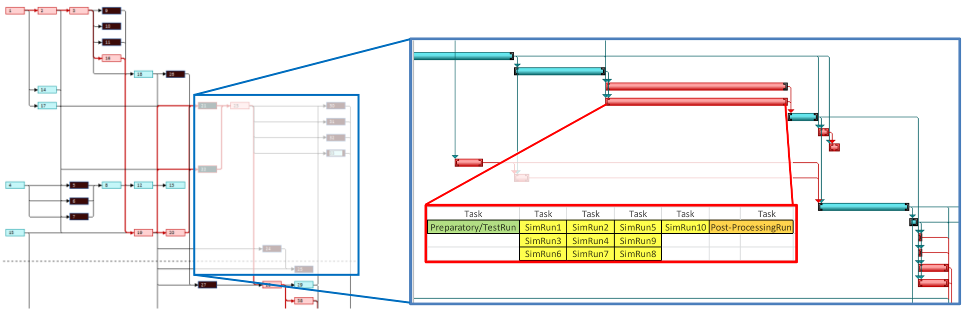


Figure 3. From a PERT graph of tasks (left) to the Gantt chart (right) and to the detailed Simulations Schedule (middle).

The design of a product is a process involving many tasks human, physical or simulations. A first step is to locate all the interconnected simulation/computation tasks in the higher level workflow and create the corresponding simulation workflow. A next step is to define the required tools such on hardware and software either local or remote, and, a last step is to locate the available resources, time & money that will pose several constrains on the scheduling process. Our goal is to use the lowest amount of resources, time & money without reducing the quality of the results. These objectives are contradictory and they make our scheduling problem harder to solve.

Workflow scheduling focuses on mapping and managing the execution of inter-dependent tasks on diverse utility services. For the special case of cloud-oriented workflow systems, one of the most important missions is to dispatch tasks to resources based on customer's requirements and the characteristics of cloud-oriented workflow tasks, as well as time-cost of scheduling [7].

In general, the problem of mapping tasks on distributed services belongs to a class of problems known as “NP hard problem”. For such problems, no known algorithms are able to generate the optimal solution within polynomial time. Although the workflow scheduling problem can be solved by using exhaustive search, the complexity of the methods for solving it is very large [8]. For grid computing environments, scheduling decision must be produced in the shortest time possible, because there are many users compute for resources and desired time slots could be occupied by others at any time. Genetic algorithms (GAs) provide robust search techniques that allow a high-quality solution to be derived from a large search space in polynomial time, by applying the principle of evolution. A successful application of GAs for workflow optimization in the aerospace manufacturing domain is also presented in [9].

3. Evolutionary and Genetic Algorithms

The Genetic Algorithm is a class of Evolutionary Algorithms that works on the principle of survival of the fittest via natural selection [10]. It combines the exploitation of best solutions from past searches with the exploration of new regions of the solution space. Any solution in the search space of the problem is represented by a member of the population (an individual – in chromosomes). The population of individuals evolves over generations. The quality of a member of the population is a score or penalty determined by a fitness-function. The score value indicates how good the individual is compared to others in the population.

A Genetic Algorithm performs the following steps:

1. Generates an initial population.
2. Computes the fitness for each individual.
3. Selects the parent couples
4. Creates the kids from the parents.
5. Selects the final members of the next generation
6. Returns to step 2 until a satisfactory solution is obtained.

The GA optimizer can use various forms of selection, cross-over/ mutation (steps 3 to 5) to evolve the initial population. The important parameters of a GA are the: Population Size, Number of Generations, Crossover/Mutation types & rates and Selection procedures, where:

- Crossover, is an exchange of substrings denoting chromosomes, for an optimization problem,
- Mutation, is the modification of bit strings in a single individual, and
- Selection is the evaluation of the fitness criterion to choose which individuals from a population will go on to reproduce.

As shown in figure 4, the GA cycle (steps 2 to 6) is repeated until a termination condition has been reached, such as: a solution that meets the criteria, the maximum number of generations, the maximum time allowed, etc. The member of the last generation with the highest score(s) is the best solution and may be accompanied by the other top candidates to create a set of best solutions proposed by the algorithm.

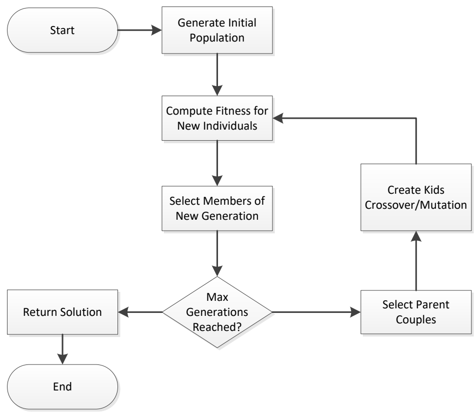


Figure 4. The Genetic Algorithm flowchart.

4. Simulation Workflow Optimization

During a product design & development process we need to perform numerous simulations and, proper scheduling of the simulations can save time, money, human or computer resources and provide better results. The proposed method is trying to solve the more complex and more realistic case of optimizing a set of simulation tasks over a set of heterogeneous computing resources. This is one step ahead of the widely studied case of multiple identical or uniform resources, like a cloud or grid computing infrastructure. In our case, the designer needs to execute a series of computational tasks, in order to investigate as many as possible solutions before concluding to the final design. Submitting all computational tasks to only one resource (e.g., grid computing) is an easy and trivial case where no optimization is required by the designer, and where the grid provider may perform a local optimization of its processors, only to improve the grid efficiency.

In this work we face the common and realistic case where the designer has a number of different computing resources available, and, he is also responsible to optimize their use. The available infrastructure may be local computing resources such as: a workstation, a dedicated server, or a server farm, or, they may be remote and rented on demand such as: a super computer, a cloud service, or a specific high performance computing (HPC) web service. Moreover the designer needs to simulate & analyze several product models using different software tools that may or may not run on each available resource.

(a)

TASK category	Task type (SW)	Duration(h)	Runs	Task Instances	SW Required	Precedence
T1/1 - FEM	Linear Finite Element Analysis	2	50	T1/1 (1 ... 20)	1	-
T1/2 - FEM	Non-Linear Finite Element Analysis	8	10	T1/2 (1 ... 10)	2	-
T1/3 - FEM	Stress Analysis	3	5	T1/3 (1 ... 5)	1;2	-
T1/4 - FEM	Deformation/Displacement Calc.	5	5	T1/4 (1 ... 5)	1;2	-
T2/1 - CFD	Comp. Fluid Dynamics Analysis	12	4	T2/1 (1 ... 4)	3	T1/1 & T1/2
T2/2 - CFD	Air Flow Simulations	14	3	T2/2 (1 ... 3)	3	T1/3
T2/3 - CFD	Noise & Pressure Calc.	15	3	T2/3 (1 ... 3)	3	T1/1 & T1/4
T3/1 - CATIA	CATIA DMU Analysis	9	5	T3/1 (1 ... 5)	4	T2/1 & T2/2
T3/2 - CATIA	CATIA Ergonomics Analysis	10	5	T3/2 (1 ... 5)	4	T2/3

(b)

Comp.Resource	Availability	Speed	Cost	Operator
Work-Station1	100	1	1	1
Local Server1	50	3	2	1
Server farm	50	10	5	0
Super-Computer	10	20	20	0
Cloud Computing	100	10	10	1

Figure 5. Sample lists of (a) Computational Tasks, and, (b) Available Resources

The proposed method solves the problem by optimizing the schedule of an intermediate simulation workflow (at a lower, but not the lowest – grid – level). This workflow contains all computational tasks or subtasks and their dependencies in detail. All costs and constraints associated with simulations are also considered as variable, enabling further optimization. Also simulation criteria such as accuracy can be varied to investigate for the optimal balance between e.g. acceptable accuracy, computational cost and simulation duration that fit the higher-level schedule. This is achieved by utilizing semantic annotations of each model and simulation task. The relationships, requirements, costs and constraints of the required simulations and the overall optimization loop are examined in detail by accessing and assessing the information available (figure 5) from the semantic annotations of each model.

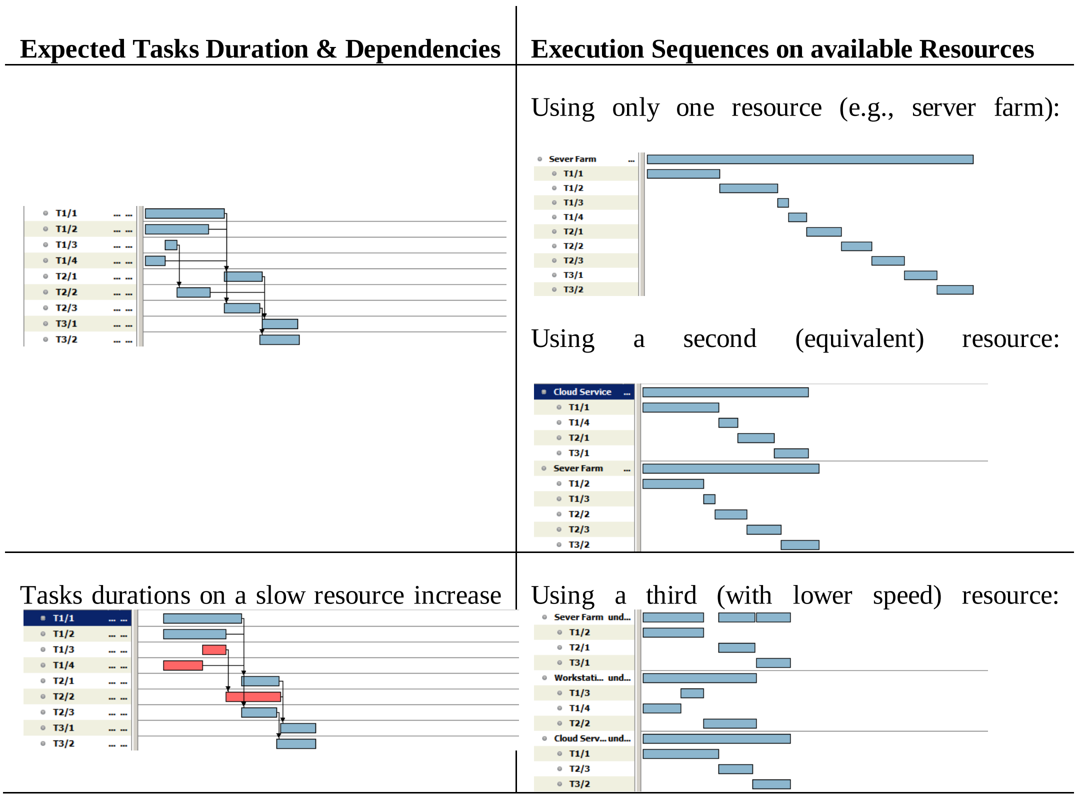
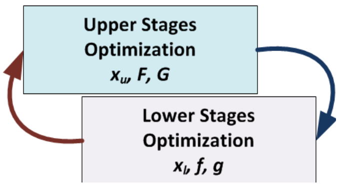


Figure 6. Simulation workflow execution time optimization using heterogeneous computational resources

When the optimization process is invoked, the Genetic Algorithm starts with an initial population of solutions which are members of the search space. The search space contains all possible combinations of task assignments on the available resources (figure 6). Each solution (schedule) is evaluated, and an overall score/penalty is assigned. Constrains violations are excluded or penalized by a quadratic function depending the severity. Schedules are sorted by their efficiency and the fittest of the last generation is considered as the best solution.

The advantage of the method is that it is situated between two other optimization stages: the optimization of the entire process higher level workflow (containing all human, physical and computational tasks) and the lowest local resource optimization level performed by the service provider. The various workflow levels create a multi-level hierarchical system and a bi-level optimization may be applied until equilibrium is reached. The simulation workflow optimization method updates the computational tasks details and informs the higher level optimization tool in order to produce an updated solution. The new task constrains are again fed into the simulation workflow optimization to re-evaluate the optimal schedule and resubmit the results (figure 7).



$$\min_{x_u, x_l} \quad F(x_u, x_l)$$
such that
$$x_l \in \operatorname{argmin}_{x_l}\{\, f(x_u, x_l) : g_j(x_u, x_l) \le 0, j = 1, \ldots, J \,\}$$
$$G_k(x_u, x_l) \le 0, k = 1, \ldots, K$$
$$x_u \in X_U, x_l \in X_L$$

Figure 7. Bi-level optimization for a multi-level hierarchical system

Another advantage of the method is its capability for on-line optimization. Whenever a change is observed on a resource characteristic, such as: availability, speed, cost, etc., the optimization algorithm is executed again either automatically or on a user demand. The revised schedule is optimal and may replace the original but suboptimal (currently) schedule. The revision, if significant enough, may also trigger a new global optimization on a higher level.

Example Use Case

The above described method has been applied for the optimization of a product development process in the aerospace manufacturing domain. Developing an aircraft part is a multiphase process starting with a proposal phase and followed by a preliminary and a detailed design phase. In each phase models are used with different level of fidelity. The challenge in each of these phases is the iterative nature between design, stress analysis, sizing and performance analysis. After a design is made, linear and non-linear stress analysis is performed on the design to check if stress limits are not exceeded. Based on that analysis the structural elements are sized in order to stay within limits. Based on the sizing a performance analysis can be made with respect to weight and cost. Another challenge lies in the use of a common geometric model of the aircraft part, usually by means of a CAD model. This CAD model is used for Finite Element Analyses and also provides information for the recurring and non-recurring cost estimations. It is extremely important that all disciplines use the same basis. Currently the design of a rudder is time-consuming and costly. It requires a lead time in the order of several months. Also time and resource constraints do not allow for many design iterations, such that a feasible but sub-optimal design is usually achieved. The proposed method for simulation workflow optimization was able to improve the search for an optimal rudder design, through investigation of multiple repeated simulations of varying configurations, while trying to satisfy conflicting requirements such as: customer satisfaction, weight minimization, manufacturing cost minimization. By optimizing each simulation workflow, a significant amount of time could be saved, or alternatively, a larger set of product variations could be tested, both resulting to increased customer satisfaction [11].

5. Conclusion

In this work an evolutionary method was presented for optimizing the execution of simulation workflows. The method is based on a heuristic optimization technique (Genetic Algorithms) that is applied to deliver an optimized workflow implementation of an initial workflow schedule. The method can be used on-line and in conjunction with a global optimization technique to continuously update the execution schedules of the computational tasks of any complex process. The proposed method demonstrates its potentials when the product design problem requires a lot of time consuming simulations or finite elements analysis under a constrained availability of computer resources.

References

[1] webmaster@ftb.ca.gov (27 October 2009). "Business Process Management Center of Excellence Glossary" (PDF). Retrieved 31 March 2015.

[2] Simtech Cluster of Excellence, University of Stuttgart, http://www.iaas.uni-stuttgart.de/forschung/projects/simtech/sim-workflows.php

[3] Tsahalis, J., Tsahalis, H.-T. Moussas, V.C. (2012) “Modeling the Comfort of Aircraft Passengers as part of the Passenger Cabin Environmental Control System (ECS)”, Proceedings of the 5th IC-SCCE, Athens, 4-7 July, 2012.

[4] Lee H. and Kim S.-S. (2001), “Integration of Process Planning and Scheduling Using Simulation Based Genetic Algorithms”. Int J Adv Manuf Technol 18:586–590, 2001.

[5] Tsahalis, J., Moussas, V.C., Tsahalis, H.-T. (2014) “An Algorithm for Distributed Heterogeneous Simulation Workflow Optimization”, Proceedings of the 6th IC-SCCE, Athens, 9-12 July, 2014.

[6] Optimus®, Process Optimization, Noesis Solutions http://www.noesissolutions.com/Noesis/. Retrieved 31 March 2015.

[7] Chunlai Chai, (2013). “Modeling Resource-Constrained Project Scheduling Problem and its Solution by Genetic Algorithm”. Journal of Digital Information Management, 11(2), 2013, pp.87-92.

[8] Jia Yu and Rajkumar Buyya, (2006) “A Budget Constrained Scheduling of Workflow Applications on Utility Grids using Genetic Algorithms” Workshop on Workflows in Support of Large-Scale Science, Proceedings of the 15th IEEE International Symposium on High Performance Distributed Computing (HPDC).

[9] Tsahalis, J., Tsahalis, H.-T., Moussas, V.C. (2013) “Optimization of a Heterogeneous Simulations Workflow”, Proceedings of the 5th IC-EpsMsO, Athens, 3-6 July, 2013

[10] Whitley L. Darrell (Ed) (1993), “Foundations of Genetic Algorithms”, MorganKaufmann 1993.

[11] Moussas, V.C., Tsahalis, J., Tsahalis, H.-T. (2014) “Simulation Workflow Optimization Application in Aerospace Manufacturing”, Proceedings of the 6th IC-SCCE, Athens, 9-12 July, 2014.