Resource allocation problem in a distributed real-time simulation platform

Ahmed Khassiba[1]

TEC, Capgemini Engineering, Sud-Ouest, 4 avenue Didier Daurat, Blagnac, France
ahmed.khassiba@capgemini.com

Keywords : *resource allocation, integer programming, graph coloring.*

1 Introduction

The development of complex engineering systems such as aircraft goes through a number of tests performed on various types of simulators depending upon the phase of the development. RT-SIM research project aims at developing tools and methods to accelerate the configuration and deployment of distributed real-time simulation platforms, so as to satisfy requests for simulation, typically expressed by system engineers. We speak of a *distributed* simulation platform, when several physical computational resources (e.g., workstations, single-board computers, microprocessors, etc) used for simulation are connected through a network. The *real-time* feature refers to the ability of the simulation platform to execute appropriate simulation modules with a given accurate time frequency. A period corresponding to this time frequency is typically expressed in the order of milliseconds or microseconds. In this work, we focus on the optimization of computational resource allocation in a distributed real-time simulation platform. We introduce the optimization problem and draw an analogy with the *list-coloring problem* in a graph with a specific structure. We propose an integer linear programming formulation, that we strengthen with several valid inequalities. We compare our formulation to two integer linear programming formulations from the literature of the list-coloring problem, and to a constraint-programming formulation.

2 Problem statement

We are given a generic simulation platform consisting of $R > 0$ computational resources, among them $R^{RT} \in [0, R]$ resources that support real-time computation, such as single-board computers (e.g., RaspBerry Pi). We assume that all resources are interconnected, such that any pair of resources can have a direct communication link. In the context of a simulation request, we are provided with a logical architecture of a simulation, made of functions and their logical links. Also, we are given $M > 0$ executable simulation modules, each corresponding to a function in the logical architecture, with $M^{RT} \in [0, M]$ modules requiring to be executed in real time. For convenience, we may call such modules, as *RT modules*, and those not requiring an accurate real-time execution, as *non-RT modules*. Moreover, we are given a compatibility matrix $A = (a_{rm})_{r\in\mathcal{R}, m\in\mathcal{M}}$, such that $a_{rm} = 1$, if resource r is compatible with module m, and $a_{rm} = 0$ otherwise. We assume that a computational resource can execute multiple simulation modules in parallel. However, to run a given simulation module with real-time constraints, a compatible computational resource should be dedicated to execute only that module of interest.

The optimization problem of computational resources allocation in a distributed real-time simulation platform (RA-DRTP) seeks to assign every simulation module to exactly one resource, so as to minimize the number of used resources, while satisfying compatibility constraints, and real-time computing constraints.

3 Complexity

The resource allocation problem in a distributed real-time simulation platform (RA-DRTP) can be reduced to an instance of the list-coloring problem. The reduction draws as follows. Each node corresponds to a simulation module. A color corresponds to a computational resource. The list of eligible colors for a given node corresponds to the list of resources compatible with the module represented by that node. An arc is drawn between every node corresponding to an RT module and every other node. Finding an allocation of resources to the simulation modules, in the sense of RA-DRTP, corresponds to allocating colors to the graph's nodes so that no two adjacent nodes have the same color, while nodes can only take colors from their list of eligible ones. Minimizing the number of used resources corresponds to minimizing the number of used colors. The complexity of the resource allocation problem (RA-DRTP) can be deduced from this reduction, as given by Theorem 1.

Theorem 1 *If $M = M^{RT}$ or $M = M^{RT} + 1$, the resource allocation problem RA-DRTP can be solved in polynomial time. Otherwise, the resource allocation problem RA-DRTP is NP-complete.*

Proof : In the general case, the graph corresponding to our problem is a *threshold* graph. The list-coloring problem in a threshold graph is known to be NP-complete [1]. In the two particular cases, where $M = M^{RT}$ or $M = M^{RT} + 1$, the graph is complete, and the list-coloring problem in a complete graph can be solved in polynomial time [1]. □

4 Mathematical formulations

We propose an integer linear programming formulation, that we tighten by using a lower-bound constraint on the objective function, and three types of valid inequalities. Two integer linear formulations are adapted from the literature of the list-coloring problem [2]: the standard, and the stable-set-based formulation. A constraint-programming formulation is adapted from the literature of coloring problem. A computational study, using CPLEX and CP Optimizer, is conducted on a large instance set to compare the performance of all formulations.

5 Results and perspectives

The tested constraint-programming formulation reveals to be the least performing. Our proposed formulation outperforms the standard formulation, while the stable-set formulation is the best performing one. Currently, a tree-search-based heuristic algorithm is being developed. Also, a bi-objective variant is being explored, where two objectives are optimized in lexicographic order. First, the number of used resources is minimized, then the resources' load (*i.e.*, the number of models assigned per resource) is balanced. Future work will focus on modeling additional realistic aspects involved in resource allocation, such as computational resource capacity and module execution periodicity.

References

[1] Flavia Bonomo, Guillermo Durán, and Javier Marenco. Exploring the complexity boundary between coloring and list-coloring. *Annals of Operations Research*, 169(1):3–16, July 2008.

[2] Mauro Lucci, Graciela Nasini, and Daniel Severín. A branch and price algorithm for list coloring problem. *Electronic Notes in Theoretical Computer Science*, 346:613–624, 2019.