Quadratic Integer Programming Approach for Reliability Optimization of Cyber-Physical Systems Under Uncertainty Theory

Conference Paper · March 2021

DOI: 10.1007/978-3-030-71017-0_58

Hassan Reza
University of North Dakota

Quadratic Integer Programming Approach for Reliability Optimization of Cyber-Physical Systems under Uncertainty Theory

Amrita Chatterjee and Hassan Reza

School of Electrical Engineering and Computer Science
University of North Dakota, Grand Forks, ND 58202, USA
amrita.chatterjee@und.edu, hassan.reza@und.edu,

Abstract. Cyber-physical systems (CPS) are an example of software and hardware components working in symphony. The greatest challenge in CPS design and verification is to design a CPS to be reliable while encountering various uncertainties from the environment and its constituent subsystems. Cost, delay and reliability of a CPS are functions of software-hardware partitioning of the CPS design. Hence, one of the key challenges in CPS design is to achieve reliability maximization while factoring in uncertainty in cost and delay .

This work leverages the problem formulation developed in recent research [13], which poses CPS design as an optimization problem for reliability assurance while factoring in uncertainty in cost and delay. In this formulation cost and delay are modeled as variables with uncertainty distributions under uncertainty theory, and the reliability requirement becomes an optimization objective. Authors of [13] also show heuristic solutions of this optimization problem can produce hardware/software partitioning which has potential to offer greater reliability under uncertainty.

The novel contribution of this work is the exploration of alternate heuristics to genetic algorithm used in [13] to solve the optimization problem. We conclude that treating the optimization problem as a 0-1 integer quadratic programming problem is feasible and then explore a few heuristics to solve such problems. Next, we solve this problem with an heuristic method. Preliminary results suggest that this solution method can achieve better reliability.

Keywords: CPS design, reliability, uncertainty theory, optimization, heuristic

1 Introduction

Cyber physical systems are composed of both hardware and software components. Apart from integration of these two components its comprised of elements like actuators, sensors, processors [10], [23], [11]. A cloud edge computing framework including the design of CPS has been put forward in [29]. Data processing and its various challenges faced have been discussed in [28] while exploring a systematic big data-as-service framework for CPS.

Among the challenges present in CPS design the most important one is to make the system reliable. In our day to day life there are lot of uncertainties related to environment, weather and other disturbances. Finding a software/hardware partition of a CPS is an ongoing challenge. Automation of this problem has been already researched and exact partitioning [5], [22], [20] or heuristic partitioning models have been deployed. [30], [6], [7]. However few algorithms work under the real world constraint that it is impossible to estimate the time and cost of the components of the system accurately. Also analyzing the attribute of reliability is intricately connected to the hardware/software partition during design of the CPS. Unless these challenges are addressed, existing partitioning algorithms are limited to its applications in the real world systems. Therefore various methods and proposals are on the rise to bridge this gap. To achieve this an uncertain programming model has been developed recently in [13] which characterizes reliability and includes uncertainty compliance.

In [13] the partitioning problem has been depicted as a mathematical optimization equation. Cost related objectives and the delay related constraints are stated in terms of uncertain variables. The characteristics of reliability are depicted as task graph of the system. This design has higher degree of assurance in terms of safety and security under the circumstances of uncertainties. The authors of [13] attempt to solve the formulated optimization problem with a simulated annealing heuristic implemented with genetic algorithm.

As a novel contribution of this work We explore alternative methods to solve the optimization problem: we pose it as a 0-1 integer quadratic programming problem which is known to be NP-hard [4] and explore heuristic methods to solve this problem. We settle on a heuristic method to solve this problem. Preliminary results indicate that solutions with improved values of the reliability metric compared to genetic algorithm solution are feasible.

2 Related Work

The existing models and algorithms for partitioning can be broadly differentiated as exact partitioning and heuristic partitioning models. The exact algorithms comprised branch-and bound [5], integer linear programming [22] and dynamic programming [16], [20], whereas the heuristic algorithm consists of simulated annealing [9], [21], [27], [26] and genetic algorithm.

These algorithms work at a smoother pace when put in their co-design environments. The parameters of all the components are deterministic, meaning for some specific inputs, it will give definite outcomes that are already predefined. However, in the design phase, the cost and time of the software components remain hard to estimate. Some works assume these factors as subjective probabilities, and make use of this assumption to perform system-level partitioning [1]. However, some of these assumptions do not hold when we consider the system as a whole from the project management view point.

Reliability has been taken to be the prioritized attribute in terms of partitioning according to [25], [14], [12]. The imprecise quantities presented in [17] are devoid of fuzziness or randomness. Therefore, based on some preliminary idea in [15], system partitioning is conducted, taking into consideration the reliability factor. Belief degree of uncertain events are measured relying on the uncertainty theory [17], [18], [2].

3 Uncertainty Model

A formal exhibition of the software/hardware partitioning issue encountered during cyber-physical system is provided in this section. This formulation including the various metrics and optimization problem definition is obtained from the extensive body of work spanning many years from Jiang et. al. [15], [13]. This formulation targets to reduce system cost and execution time while enhancing reliability under uncertainty theory.

3.1 Problem Setup

The cyber-physical system under design is modeled as a directed acyclic graph $G(V, E)$, where V is the set of nodes $\{v_1, v_2, ..., v_n\}$ and E is the set of edges $\{e_{ij}|1 \leq i < j \leq n\}$

1. γ_i^h denotes the additional cost of the hardware implementation of node i in the system, over a software implementation.
2. δ_{ci}^h denotes the linear uncertainty distribution of γ_i^h, denoted by $\lambda(p_{ci}^h, q_{ci}^h)$ where p_{ci}^h, q_{ci}^h are non-negative real numbers.
3. t_i^h denotes the execution time of node i if implemented in hardware while, t_i^s denotes the execution time of an equivalent software implementation.
4. δ_{ti}^h, δ_{ti}^s are uncertainty distributions of uncertain variables t_i^h, t_i^s respectively, denoted by $\lambda(p_{ti}^h, q_{ti}^h)$, $\lambda(p_{ti}^s, q_{ti}^s)$, where p_{ti}^h, q_{ti}^h, p_{ti}^h, q_{ti}^h are non-negative real numbers.
5. c_{ij} is the communication time between nodes i and j, it is given a non-zero value only if nodes i and j differ in implementation choice.
6. r_i^h denotes the reliability of node i if implemented in hardware while, r_i^s denotes the reliability of an equivalent software implementation.

The hardware software partitioning problem is defined as finding a bipartition P of this graph, where $P = (V_h, V_s)$ such that $V_h \cup V_s = V$ and $V_h \cap V_s = \delta$. In addition T_0 is the given execution time upper bound while R_0 is the given reliability lower bound. We can represent such a partition in terms of a n-wide vector of binary index variables $x = \{x_1, x_2, ..., x_n\}$. If node i is implemented in software then it is assigned a value of 1, else it is assigned a value 0. The objective is to find a partition P such that $T(x) \leq T_0$ and $R(x) \geq R_0$, while total cost $H(x)$ is minimized.

3.2 Optimization Problem Metrics

Evaluation of a partition is based on three primary metrics, reliability, execution time and cost discussed in [13]. The total cost includes both software and hardware components of the CPS. Similarly, the execution time is sum of the execution time of each node and the communication time between nodes. The reliability is assumed to be the probability that the system will perform its intended function accurately, which requires all nodes to function appropriately.

Cost Metric The total cost of the system can be represented by sum of the additional cost of the hardware implemented nodes (with an inherent assumption that hardware implementation is more costly than software implementation) for the purposes of optimization. Hence cost $H(x)$ of the partition x can be formulated as:

$$H(x) = \sum_{i=1}^{n} \gamma_i^h (1 - x_i) \tag{1}$$

Time Metric Time metric is addition of two parts: execution time of each node and communication time between nodes. Again, it is a rational assumption that two nodes that do not alter in implementation choice will have zero communication cost. Total time is sum of total execution time and total communication time, which is expressed as follows

$$T(x) = \left[\sum_{i=1}^{n} t_i^s x_i + t_i^h (1 - x_i)\right] + \left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} [(x_i - x_j)^2]\right] \tag{2}$$

Reliability Metric Nodes that are devoid of any outgoing edges are the destination / system output nodes, while nodes that have zero incoming edges are regarded as start / system input nodes. The adjacency matrix $Adj[n][n]$ represents the dependency edges of parent to child nodes in the directed acyclic graph. Based on the theory of fault tree and reliability block diagram, the reliability of the task node x_i is the product of its parent nodes and reliability of itself. Hence, The system reliability $R(x)$ is the summation of the reliability of all output nodes. The reliability of each output node is obtained in a recursive manner as shown in algorithm 1.

Algorithm 1: System reliability $R(x)$

Input: $Adj[][]$ Adjacency matrix of the DAG $G(V, E)$
binary vector x
Output: $R(x)$
Init: $R(x) \leftarrow 0$

Function *Recursive(m)* **is**
  for $i \leftarrow 1$ **to** n **do**
    if *Adj[i][m] == 1* **then**
      $R_m \leftarrow R_m.(r_m^s x_m + r_m^h (1 - x_m)).Recursive(i)$
    end
  end
  return R_m
end
foreach *output node m in the DAG G(V,E)* **do**
  $R(x) \leftarrow R(x) + Recursive(m)$
end
return $R(x)$

3.3 Optimization Problem Formulation

Based on the metric definitions in previous section, the given constraint M on total execution time, and a lower bound R_0 on the reliability, the hardware/software partitioning problem is modeled as given in P_0.

$$P_0 : \begin{cases} \text{minimize} & H(\mathrm{x}) \\ \text{subject to} & T(\mathrm{x}) \leq M \\ & R(\mathrm{x}) \geq R_0 \\ & \mathrm{x} \in \{0,1\}^n \end{cases} \tag{3}$$

Since minimizing the value of $H(x)$ is equivalent to maximizing the value of $\sum_{i=1}^{n} \gamma_i^h x_i$, problem P_0 can be reduced to problem P_1.

$$P_1 : \begin{cases} \text{maximize} & \sum_{i=1}^{n} \gamma_i^h x_i \\ \text{subject to} & \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} \left[\left(x_i - x_j \right)^2 \right) \right] + \\ & \sum_{i=1}^{n} \left(t_i^s - t_i^h \right) x_i \leq M - \sum_{i=1}^{n} t_i^h \\ & 1 - R(x) \leq 1 - R_0 \\ & \mathbf{x} \in \{0,1\}^n \end{cases} \tag{4}$$

The uncertain objective function is targeted first . It is known that if $\gamma_1, \gamma_2 \cdots \gamma_n$ are uncertain variables with uncertain distributions $\delta_1, \delta_2 \cdots \delta_n$ and the function $f\left(\mathbf{x}, \gamma_1, \gamma_2 \cdots \gamma_n\right)$ is strictly increasing with respect to $(\gamma_1, \gamma_2 \cdots \gamma_m)$ and strictly decreasing with respect to $(\gamma_{m+1}, \gamma_{m+2} \cdots \gamma_n)$ Then, the converted expected objective function can be calculated as:

$$\begin{aligned} E\left[f\left(\mathrm{x}, \gamma_1, \gamma_2 \cdots \gamma_n\right)\right] = \int_0^1 f\left(\mathrm{x}, \delta_1^{-1}(\alpha), \delta_2^{-1}(\alpha)\right. \\ \left. \cdots \delta_m^{-1}(\alpha), \delta_{m+1}^{-1}(1-\alpha) \cdots \delta_n^{-1}(1-\alpha)\right) d\alpha \end{aligned} \tag{5}$$

where $\delta_i^{-1}(\alpha)$ is $(1-\alpha)\left(p_{ci}^h - q_{ci}^s\right) + \alpha\left(q_{ci}^h - p_{ci}^s\right)$. The mathematical exposition in [13] describes how this problem can eventually be converted to the following reliability maximization problem:

$$P_{final} : \begin{cases} \text{maximize} & R(x) \\ \text{subject to} & \sum_{i=1}^{n} \left(q_{ti}^s - p_{ti}^h \right) x_i + \\ & \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} \left(x_i - x_j \right)^2 \leq M - \sum_{i=1}^{n} p_{ti}^h \\ & \sum_{i=1}^{n} \left[\left(\int_0^1 \delta_i^{-1}(a) d\alpha \right) (1 - x_i) \right] \leq N \\ & x_i \in \{0,1\}; i = 1, 2 \cdots n \end{cases} \tag{6}$$

4 Research Contribution: Solution with 0-1 Quadratic Integer Programming Heuristic

The final reliability maximization problem is an instance of a multi-objective 0-1 integer quadratic programming problem, which is known to be a NP-hard [4], hence only

Table 1. Preliminary results on randomly generated DAGs

Name	Node	Edge	Gain over GA
random1	500	1,000	1.21
random2	100	1,200	1.37
random3	1,500	3,000	-0.32
random4	2,000	4,000	2.14
random5	2,500	5,000	1.79
random6	3,000	6,000	1.48

heuristic-based algorithms are applicable. Several general purpose heuristic algorithms discussed in Section 2 can be implemented to solve the resultant optimization problem. The authors of [13] chose a genetic algorithm with simulated annealing heuristics. However we realized that since this is a special case of 0-1 integer quadratic programming, more specialized heuristics such as [19] exist and can be applied to this problem.

Quadratic 0-1 integer programming is an actively explored research area. While the general problem is NP-hard, certain classes of this problem are even amenable to exact solutions [3]. However, vast majority of problem instances are only solvable with heuristic methods. We explored a number of heuristic solution methods for this problem including but not limited to genetic (GA) [19], alternating direction method of multipliers (ADMM) [24] and convex-hull based [8] heuristics.

The structure of the particular problem instance at hand lends itself to be suitable to the convex-hull heuristic (CHH) based approach [8], since its a non-linear optimization problem with linear constraints. This heuristic is based on simplical decomposition (SD), which is applied repeatedly to obtain feasible solutions for a nonlinear integer programming problem with linear constraints. For this problem SD is started each time from a different feasible point. This decomposition method allows us to use the linear programming tools such as CPLEX on each iterative step of the process.

5 Preliminary Results

Although we are still exploring the space of specialized heuristics for this problem, we implemented a heuristic algorithm similar to as described in [8], and the very preliminary experiments on random graphs show some improvement over the genetic algorithm solution. Table 1 shows the improvement over the genetic algorithm solution on the reliability metric.

6 Conclusion

This work leverages the uncertainty theory based formulation of the CPS hardware-software partition problem as developed in [13]. The resultant optimization problem is treated as a 0-1 quadratic integer programming problem and a heuristic-based solution method is applied. Preliminary results promises some improvement over genetic algorithm derived solution.

References

1. Albuquerque, J., Coelho, C., Cavalcanti, C.F., Cecilio da Silva, D., Fernandes, A.O.: System-level partitioning with uncertainty. In: Proceedings of the Seventh International Workshop on Hardware/Software Codesign, p. 198–202 (1999)
2. B. Liu: Fuzzy random dependent-chance programming. IEEE Transactions on Fuzzy Systems **9**(5), 721–726 (2001)
3. Chakradhar, S.T., Bushnell, M.L.: A solvable class of quadratic 0–1 programming. Discrete Applied Mathematics **36**(3), 233 – 251 (1992)
4. Chaovalitwongse, W.A., Androulakis, I.P., Pardalos, P.M.: Quadratic Integer Programming: Complexity and Equivalent Forms, pp. 3153–3159. Springer US (2009)
5. Chatha, K.S., Vemuri, R.: Hardware-software partitioning and pipelined scheduling of transformative applications. IEEE Transactions on Very Large Scale Integration (VLSI) Systems **10**(3), 193–208 (2002)
6. Dick, R.P., Jha, N.K.: Mogac: a multiobjective genetic algorithm for hardware-software cosynthesis of distributed embedded systems. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **17**(10), 920–935 (1998)
7. Eles, P., Peng, Z., Kuchcinski, K., Doboli, A.: System level hardware/software partitioning based on simulated annealing and tabu search. Design Automation for Embedded Systems **2**, 5–32 (1997)
8. Guignard, M., Ahlatcioglu, A.: The convex hull heuristic for nonlinear integer programming problems with linear constraints and application to quadratic 0–1 problems. Journal of Heuristics (2020)
9. Gupta, R.K., De Micheli, G.: Hardware-software cosynthesis for digital systems. IEEE Design Test of Computers **10**(3), 29–41 (1993)
10. Jeschke, S., Brecher, C., song, H., Rawat, D.B.: Industrial Internet of Things. Springer International Publishing (2017)
11. Jiang, Y., Song, H., Wang, R., Gu, M., Sun, J., Sha, L.: Data-centered runtime verification of wireless medical cyber-physical system. IEEE Transactions on Industrial Informatics **13**(4), 1900–1909 (2017)
12. Jiang, Y., Song, H., Yang, Y., Liu, H., Gu, M., Guan, Y., Sun, J., Sha, L.: Dependable model-driven development of cps: From stateflow simulation to verified implementation. ACM Trans. Cyber-Phys. Syst. **3**(1) (2018)
13. Jiang, Y., Wang, M., Jiao, X., Song, H., Kong, H., Wang, R., Liu, Y., Wang, J., Sun, J.: Uncertainty theory based reliability-centric cyber-physical system design. In: 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 208–215 (2019)
14. Jiang, Y., Wang, M., Liu, H., M. Hosseini, J.S.: Dependable integrated clinical system architecture with runtime verification. In: Proceedings of the 36th International Conference on Computer-Aided Design, p. 951–956 (2017)
15. Jiang, Y., Zhang, H., Jiao, X., Song, X., Hung, W.N.N., Gu, M., Sun, J.G.: Uncertain model and algorithm for hardware/software partitioning. 2012 IEEE Computer Society Annual Symposium on VLSI pp. 243–248 (2012)
16. Knudsen, P., Madsen, J.: Pace: A dynamic programming algorithm for hardware/software partitioning. In: CODES, pp. 85–93. IEEE Computer Society (1996)
17. Liu, B.: Uncertainty theory. In: Uncertainty Theory, pp. 1–79. Springer Berlin Heidelberg (2010)
18. Liu, B., Liu, Y.K.: Expected value of fuzzy variable and fuzzy expected value models. IEEE Transactions on Fuzzy Systems **10**(4), 445–450 (2002)

19. Lodi, A., Allemand, K., Liebling, T.M.: An evolutionary heuristic for quadratic 0-1 programming. European Journal of Operational Research **119**(3), 662–670 (1999)
20. Madsen, J., Grode, J., Knudsen, P.V., Petersen, M.E., Haxthausen, A.: Lycos: The lyngby co-synthesis system. Des. Autom. Embedded Syst. **2**(2), 195–235 (1997)
21. Niemann, R., Marwedel, P.: Hardware/software partitioning using integer programming. In: Proceedings of the 1996 European Conference on Design and Test, p. 473 (1996)
22. Niemann, R., Marwedel, P.: An algorithm for hardware/software partitioning using mixed integer linear programming. Des. Autom. Embedded Syst. **2**(2), 165–193 (1997)
23. Song, H., Rawat, D.B., Jeschke, S., Brecher, C.: Cyber-Physical Systems: Foundations, Principles and Applications, 1st edn. Academic Press, Inc. (2016)
24. Takapoui, R., Moehle, N., Boyd, S., Bemporad, A.: A simple effective heuristic for embedded mixed-integer quadratic programming (2015)
25. Tosun, S., Mansouri, N., Arvas, E., Kandemir, M., Xie, Y., Hung, W.: Reliability-centric hardware/software co-design. In: Sixth international symposium on quality electronic design (isqed'05), pp. 375–380 (2005)
26. Vahid, F., Gajski, D., Gong, J.: A binary-constraint search algorithm for minimizing hardware during hardware/software partitioning. In: Proceedings of the Conference on European Design Automation, p. 214–219 (1994)
27. Vahid, F., Gajski, D.D.: Clustering for improved system-level functional partitioning. In: Proceedings of the 8th International Symposium on System Synthesis, p. 28–35. Association for Computing Machinery, New York, NY, USA (1995)
28. Wang, X., Yang, L.T., Huazhong, L., Deen, M.J.: A big data-as-a-service framework: State-of-the-art and perspectives. IEEE Transactions on Big Data pp. 1–1 (2017)
29. Wang, X., Yang, L.T., Xie, X., Jin, J., Deen, M.J.: A cloud-edge computing framework for cyber-physical-social services. IEEE Communications Magazine **55**(11), 80–85 (2017)
30. Zhao, X., Zhang, H., Jiang, Y., Song, S., Jiao, X., Gu, M.: An effective heuristic-based approach for partitioning. Journal of Applied Mathematics (2013)