Department of
Information Engineering
and Computer Science DISI

UNIVERSITY OF TRENTO - Italy

DISI - Via Sommarive 5 - 38123 Povo - Trento (Italy)
http://www.disi.unitn.it

MULTIPROCESSOR SCHEDULING FOR REAL-TIME SYSTEMS: ERRATA

Tadeus Prastowo

September 2018

Technical Report # DISI-18-018

Multiprocessor Scheduling for Real-Time Systems: Errata

Tadeus Prastowo
DISI, University of Trento
Trento, Italy
tadeus.prastowo@unitn.it

Abstract

This paper contains a list of reference and bibliography errors found in the monograph that is titled “Multiprocessor Scheduling for Real-Time Systems”, authored by Sanjoy Baruah, Marko Bertogna, and Giorgio Buttazzo, and published by Springer International Publishing in 2015 with DOI 10.1007/978-3-319-08696-5. This paper also proposes the fixes to the errors, which we produced over the course of three working days. In total, we inspected 420 individual non-distinct references found on 129 out of the 214 pages (60%) from Chapter 1 to 23. And, out of the 174 bibliography entries in total, the references to 99 bibliography entries (57%) were fixed. The inspection of the 420 individual non-distinct references were needed because a simple find-and-replace failed to work owing to 33 out of the 99 bibliography entries (33%) being referenced inconsistently. Before applying the proposed fixes, there were 14 orphaned bibliography entries (i.e., entries with no passage referencing them). After applying the proposed fixes, all of them would have parents, while bibliography entry 87 and 88 would be orphaned. Lastly, while we have communicated our findings to the authors of the monograph, the authors have not communicated to us that the fixes that we proposed in this paper are completely correct.

I. Methodology

In identifying the errors and proposing the fixes, we matched the context of the referencing passages to, first, the titles of the referenced bibliography entries, and then, in many occasions when we were able to obtain the referenced literature, the texts of the entries themselves. We also resolved three points of doubt by corresponding directly with Prof. Baruah using e-mails.

Our endeavor was not easy because the references to 33 out of the 99 bibliography entries (33%) that we fixed were inconsistent. For example, for bibliography entry 61, the reference “... linear-time selection [61] ...” on page 141 correctly referred to entry 61, but the reference “... exhaustive-search procedure is described in [61] ...” on page 117 actually referred to entry 62. This inconsistency prevented a straightforward find-and-replace (e.g., replacing all references to entry 61 with references to entry 62), not to mention the complication that arises from citation ranges (e.g., the references “... Ha and Liu [105–107] ...” on page 25 implicitly refers to entry 106, but the usual *find* facility cannot find it). As a result, we had to peruse the individual referencing passages, which contain 420 individual non-distinct references, over a total of 129 out of the 214 pages (60%) from Chapter 1 to 23.

We note that the following bibliography entries will be orphaned after applying the proposed fixes in Section II:

- 87. Fisher, N., Baruah, S.: A polynomial-time approximation scheme for feasibility analysis in static-priority systems with bounded relative deadlines. In: Proceedings of the 13th International Conference on Real-Time Systems. Paris, France (2005).
- 88. Fisher, N., Baruah S.: Global static-priority scheduling of sporadic task systems on multiprocessor platforms. In: Proceeding of the IASTED International Conference on Parallel and Distributed Computing and Systems. IASTED, Dallas, TX (2006).

II. Results

Page 4:

- main memory ~~[161]~~[162]. Once a processing element gains access to the main memory,

Page 5:

- are active ~~[163]~~[164].

Page 8:

- Trietsch [16] and Pinedo ~~[155]~~[156]) has come up with the following classification of

Page 9:

- of the more widely-used models include the *Liu and Layland model* ~~[137]~~[138]
- *three-parameter model* ~~[132, 133, 147]~~[133, 134, 148].
- directed acyclic graph (DAG) based models have been proposed [38, ~~63~~64, ~~135~~136, ~~166~~167]

Page 10:

- (see, e.g., [52, ~~74~~75, ~~101~~102, ~~122~~123, ~~146~~147, ~~169~~170]); however, this body of results is dwarfed by

Page 11:

- e.g., ~~[65, 108]~~[66, 109]), this work is not really mature enough for us

Page 14:

- This task model was formally defined in a paper ~~[139]~~[138] coauthored by C. L. Liu and

Page 15:

- indicated by the name of the model, each task in this model ~~[149]~~[148] is characterized by

Page 18:

- systems such as Linux, some work has recently been done [39, ~~104~~103] on a migrative

Page 19:

- precisely and methodically demarcated in Stigge's dissertation ~~[168]~~[167]; see also ~~[169]~~[168]).
- sporadic behavior is sometimes brushed aside (see, e.g., [~~140~~139, p. 40], which claims

Page 22:

- is to be found in the survey paper ~~[169]~~[168] and in Stigge's dissertation ~~[168]~~[167].

Page 25:

- monotonic scheduling algorithm ~~[139]~~[138], which assigns priorities to tasks according
- The earliest deadline first scheduling algorithm [76, ~~139~~138], in which the
- the Least Laxity algorithm discussed in Sect. 20.1 and EDZL ~~[125]~~[124] discussed in
- were studied by Ha and Liu ~~[105–107]~~[104–106], who proposed the following definition:
- **Definition 3.3 (priority-driven algorithms ~~[107]~~[106].)**
- designed (see, e.g., ~~[150]~~[149]).

Page 28:

- However, Ha and Liu proved ~~[107]~~[106] that *all preemptable FTP and FJP scheduling*
- Priority-driven algorithms were defined in ~~[107]~~[106];
- term "predictability" as used here is also from ~~[107]~~[106].

Page 30:

- it has been shown ~~[112, 124]~~[113, 125] that the smallest fix-point of the recurrence

Page 31:

- The rate-monotonic (RM) scheduling algorithm ~~[137]~~[138] is an FTP scheduling
- known ~~[137]~~[138] that RM is an optimal FTP scheduling algorithm for scheduling
- The DM scheduling algorithm ~~[133]~~[134] is another FTP scheduling algorithm in which
- It has been shown ~~[133]~~[134] that DM is an

Page 32:

- of SCP, with the following parameters [49, ~~132~~133].

Page 33:

- It is evident (see ~~[132]~~[133] for details) that this task system is feasible if and only if there
- Two important results were obtained in ~~[133]~~[134] concerning fixed-task-priority (FTP)

Page 35:

- Recall that an *implicit-deadline sporadic task* ~~[137]~~[138], also

Page 36:

- the rate-monotonic (RM) utilization bound of ln 2 (approx. 0.69; ~~[137]~~[138]) are among

Page 39:

- scheduling algorithm [76, ~~139~~138];
- concerning fixed-task-priority scheduling. It follows from the results in ~~[139]~~[138]
- It is widely known that such partitioning is equivalent to the bin-packing [~~112~~111,
- ~~113~~112] problem, and is hence highly intractable:

Page 40:

- Lopez et al. ~~[142]~~[141] compared several widely used heuristic algorithms extensively.
- In making these comparisons, Lopez et al. ~~[142]~~[141] classified the studied heuristics
- **Definition 6.1** (from ~~[142]~~[141])
- All the heuristic algorithms considered by Lopez et al. ~~[142]~~[141] were RA ones
- a non-RA partitioning algorithm. The algorithms considered by Lopez et al. ~~[142]~~[141]

Page 41:

- Lopez et al. ~~[142]~~[141] compared these different partitioning algorithms from the
- We now briefly list some of the results obtained by Lopez et al. ~~[142]~~[141]. Suppose we
- obtained in ~~[142]~~[141] on the utilization bounds of any reasonable allocation algorithm.
- First, it was shown [~~142~~141, Theorem 1] that any reasonable allocation algorithm has
- The following lower bound was also proved in [~~142~~141, Theorem 2]: No allocation

Page 42:

- The following utilization bounds were shown for specific algorithms in ~~[142]~~[141]:
- Lopez et al. ~~[142]~~[141] can be implemented extremely efficiently:

Page 43:

- computed in ~~[142]~~[141]. But it is quite evident that while
- (including the RA and RAD ones studied by Lopez et al. ~~[142]~~[141]).
- while the speedup factors of the remaining six algorithms considered in ~~[142]~~[141]
- alone cannot explain experimentally observed conclusions (see, in, e.g, [~~144~~143;

Page 44:

- Hochbaum and Shmoys ~~[108]~~[107] designed a PTAS
- then the algorithm in ~~[108]~~[107] will, in-time polynomial in the representation
- as a *resource augmentation* result ~~[116]~~[115]: the algorithm of ~~[108]~~[107] can partition, in
- by an optimal algorithm, provided it (the algorithm of ~~[106]~~[107]) is given augmented
- However, the algorithm of ~~[108]~~[107] has poor implementation efficiency in practice:
- The ideas in ~~[108]~~[107] were applied in [68] to obtain an implementation that is efficient
- Actually, the result in ~~[108]~~[107] was expressed in terms of minimizing the *makespan*
- problem considered in ~~[108]~~[107] is easily shown to essentially be equivalent to the problem

Page 50:

- the optimality of EDF on preemptive uniprocessor platforms [76, ~~139~~138] that

Page 51:

- As we had stated in Sect. 4.2, Liu and Layland ~~[139]~~[138] showed that the FTP priority
- and that the following property of RM was proved in ~~[139]~~[138]:
- **Theorem 6.3** *(from ~~[139]~~[138])*
- Oh and Baker ~~[153]~~[152] applied this uniprocessor utilization bound test to partitioned
- Diaz and Garcia ~~[141]~~[140] refined and generalized this result
- **Theorem 6.4** *(from ~~[141]~~[140])*
- Guan et al. ~~[101, 102]~~[100, 101] have devised a semi-partitioned
- to ~~[101, 102]~~[100, 101].

Page 52:

- is from [68]; which applied the results in ~~[108]~~[107] concerning makespan minimization

Page 53:

- The following result was established by Horn ~~[107]~~[108]:
- **Theorem 7.1** (from ~~[107]~~[108])
- by means of the technique of Coffman and Denning [~~79~~80;
- Here is an informal description of the technique; for further detail, please see [~~79~~80, p. 116].

Page 60:

- flow theorem of Ford and Fulkerson ~~[90]~~[91]

Page 61:

- the integer flow theorem ~~[90]~~[91], the existence

Page 65:

- e.g., ~~[173]~~[174] defined a variant called boundary fair (BF) that only enforces
- concepts called DP-FAIR was recently proposed in ~~[134]~~[135]

Page 66:

- (RUN) algorithm ~~[159]~~[160] is another efficient implementation that seeks to
- see, e.g., the dissertation by Srinivasan ~~[165]~~[166].

Page 67:

- was obtained by Phillips et al. ~~[153]~~[154], who explored the use of
- Phillips et al. ~~[153]~~[154] was the scheduling of real-time systems that could be modeled
- as collections of independent jobs, not recurrent tasks. In ~~[153]~~[154], it was shown that
- algorithms was formally proposed by Kalyanasundaram and Pruhs ~~[113]~~[114].

Page 68:

- **Theorem 8.1** *(from ~~[153]~~[154])*
- In this chapter, we will describe how the techniques introduced in ~~[153]~~[154] have been

Page 71:

- from ~~[153]~~[154], is somewhat tedious and detailed. It is presented here for the sake of

Page 73:

- processors, which was originally proved in ~~[153]~~[154], can be derived as an immediate

Page 76:

- Phillips et al. ~~[153]~~[154] had proved that any instance of jobs feasible upon

Page 77:

- which is commonly called the *Dhall effect* ~~[78]~~[79]

Page 78:

- Very efficient implementations of EDF have been designed (see, e.g., ~~[148]~~[149]).

Page 80:

- optimality of EDF on uniprocessors ~~[137]~~[138].

Page 82:

- circumvent the Dhall effect ~~[78]~~[79]

Page 85:

- Many of the results presented in this chapter build upon the ideas in ~~[153]~~[154].
- are from ~~[91, 92]~~[92, 93]. The utilization bound for EDF was derived in ~~[95]~~[96]; Algorithm PRID

Page 87:

- It has long been known ~~[137]~~[138] that the rate monotonic (RM) scheduling
- RM, too, suffers from the Dhall effect ~~[78]~~[79];

Page 90:

- A better utilization bound for global RM was proved in ~~[59]~~[60]:
- **Theorem 9.3** *(from ~~[59]~~[60])*

Page 91:

- Recall from Sect. 8.3, that the *Dhall effect* ~~[78]~~[79] (see Example 8.1)

Page 92:

- An improvement to the above utilization bound was presented in ~~[59]~~[60];
- **Theorem 9.5** *(Corollary 2 in ~~[59]~~[60])*

Page 93:

- This bound was obtained by adapting ideas from ~~[137]~~[138] to construct a task system
- improved utilization bound in Theorem 9.5 is from ~~[59]~~[60].

Page 96:

- Recall that a *3-parameter sporadic task* (~~[147]~~[148]; henceforth often

Page 97:

- called *Liu & Layland task systems*, since they were popularized in a paper ~~[137]~~[138]

Page 98:

- it was observed [24, ~~62~~63] that it was useful to consider some

Page 99:

- It follows from results in ~~[80]~~[81] that
- pseudo-polynomial times algorithms are known for computing DBF [51, ~~160~~161].
- accuracy have also been designed [26, ~~85~~86, ~~88~~].
  Note on the removal of the reference to entry 88: To quote the referencing passage, "Polynomial-time algorithms for computing DBF *approximately to any desired degree of accuracy have also been designed". We first thought that the*

reference to entry 88 would be best fixed by repointing it to entry 87. However, in the text of the bibliography entry 87, Fisher and Baruah designed a polynomial-time algorithm for computing RBF, *not* DBF. *Prof. Baruah informed us by e-mail that this was indeed the case and suggested that the reference to entry 88 should not have been referenced.*

Page 100:

- It was observed [24, ~~62~~63] that some jobs arriving and/or having deadlines outside
- *demand* ~~[62]~~[63].

Page 101:

- The following function RBF* approximating RBF has been proposed ~~[86]~~[89]:

Page 102:

- such approximations may be found in Fisher's dissertation ~~[83]~~[84].

Page 110:

- scheduling of constrained-deadline sporadic task systems on uniprocessors ~~[133]~~[134].
- algorithm, called Algorithm FBB-FFD, was defined in ~~[88]~~[89];

Page 114:

- was studied in ~~[87]~~[89].
- dissertation ~~[83]~~[84].

Page 116:

- uniprocessors it is known [51, ~~132~~133, ~~137~~138] that there is a unique worst-case job arrival

Page 117:

- exhaustive-search procedure is described in ~~[61]~~[62]).

Page 118:

- been shown ~~[76]~~[77] that in the multiprocessor case, no algorithm can be optimal
- **Theorem 12.1** *(Dertouzos and Mok 1989 ~~[76]~~[77])*
- 3-parameter sporadic task systems, it was shown ~~[89]~~[90]
- **Theorem 12.2** *(Theorem 3 from ~~[89]~~[90])*

Page 119:

- establishing the impossibility of optimal on-line scheduling is from ~~[89]~~[90].

Page 125:

- schedulability analysis ~~[103]~~[102].)

Page 132:

- was introduced by Ha and Liu ~~[105–107]~~[104–106].

Page 142:

- explored in ~~[103]~~[102].
- details may be found in ~~[103]~~[102].

Page 164:

- Sect. 17.1.4, is from ~~[123]~~[122].

Page 170:

- This can be thought of as a generalization of the technique introduced earlier by Lawler ~~[122]~~[121] to

Page 172:

- the carry-in instances to consider, presented in Sect. 18.3, is from ~~[100]~~[122].

Page 189:

- upon multiprocessors, it was generally believed (see, e.g., ~~[133]~~[132]) that LL is strictly
- However, this was shown to not be true by Kalyanasundaram et al. ~~[117]~~[116];

Page 190:

- in the EDZL scheduling algorithm ~~[125]~~[124].
- In addition, Piao et al. proved in ~~[156]~~[155] the following utilization bound for

Page 191:

- Earlier models of this form include the moldable tasks model ~~[146]~~[145]
- the fork-join or parallel synchronous task model ~~[120]~~[119], etc.

Page 193:

- the Liu and Layland model ~~[139]~~[138] or the three-parameter model ~~[149]~~[148] in the sense

Page 194:

- in fact, it has been shown ~~[127]~~[126] to remain so even for schedulers given access

Page 195:

- The following results are derived in ~~[138]~~[137] concerning the global scheduling of
- Additionally, a *federated* scheduling algorithm is proposed and analyzed in ~~[138]~~[137].
- It was proved in ~~[138]~~[137] that this federated scheduling algorithm has a capacity

Page 204:

- federated scheduling of implicit-deadline sporadic tasks is from ~~[138]~~[137].

Page 205:

- components are called *unrelated* or *heterogeneous* ~~[145]~~[144] multiprocessor platforms.

Page 206:

- upon unrelated multiprocessors was considered in ~~[121]~~[120].

Page 207:

- an ILP exactly takes exponential time, approximation techniques ~~[111, 128, 172]~~[110, 127, 171]
- Furthermore it follows from results in ~~[128]~~[127] that under the assumption

Page 208:

- rithm ~~[119]~~[118] or the interior point algorithm ~~[118]~~[117]. (In addition, the exponential-time

Page 209:

- since ILP is also known to be intractable (NP-complete in the strong sense ~~[154]~~[153]).
- *LP-relaxation* ~~[164]~~[163] of ILP-Feas($\mathbf{U}_{[n\times m]}$):

Page 210:

- approximation techniques described in ~~[129]~~[128];
- ~~129~~128] for details.

Page 211:

- of this kind are referred to as *limited unrelated multiprocessors* ~~[173]~~[172].
- A PTAS was designed in ~~[173]~~[172] that approximately partitions systems of LL tasks
- deadline sporadic task systems, Marchetti-Spaccamela et al. ~~[147]~~[146] considered the
- by an optimal algorithm, then it can be partitioned by the algorithm in ~~[147]~~[146] upon a
- too large a factor for the algorithm in ~~[147]~~[146] to be considered practically significant
- algorithm in ~~[147]~~[146] reduces to a PTAS.

Page 213:

- upper bound on the degree of such tardiness can be provided [78, 83, ~~130~~129–
- ~~132~~131]. Several dissertations have been written on the subject of bounded-tardiness
- (TUF) ~~[160]~~[159] represent another somewhat popular approach to soft-real-time

Page 214:

- concurrent consideration of multiple resources; see, e.g., [33, 43, 85, 94, ~~151~~150, ~~152~~151,
- ~~170~~169, ~~174~~173]. There has been some work on implementing resource-sharing protocols
- Protocol ~~[166]~~[165], to multiprocessor platforms (e.g., ~~[158, 159]~~[157, 158]); and some new and
- (see, e.g., [66, ~~110~~109]).

Page 215:

- 5. Andersson, B.: Global static-priority preemptive multiprocessor scheduling with utilization bound 38%. In: Proceedings of the 12th International Conference on Principles of Distributed Systems. ~~IEEE Computer Society Press, Luxor, Egypt (2008)~~*Lecture Notes in Computer Science*, vol. 5401, pp. 73–88. Springer (2008)

Page 216:

- 23. Baker, T., Baruah, S.: Sustainable multiprocessor scheduling of sporadic task systems. In: Proceedings of the EuroMicro Conference on Real-Time Systems. IEEE Computer Society Press, Dublin ~~(2008)~~(2009)

Page 218:

- 60. Bertogna, M., Cirinei, M., Lipari, G.: New schedulability tests for real-time tasks sets scheduled by deadline monotonic on multiprocessors. In: Proceedings of the 9th International Conference on Principles of Distributed Systems.

~~IEEE Computer Society Press, Pisa, Italy (2005)~~*Lecture Notes in Computer Science*, vol. 3974, pp. 306–321. Springer (2006)

Page 219:

- 76. Dertouzos, M.: Control robotics: the procedural control of physical ~~processors~~processes. In: Proceedings of the IFIP Congress, pp. 807–813 (1974)
- 77. Dertouzos, M., Mok, A.: Multiprocessor ~~scheduling in a hard real-time environment~~on-line scheduling of hard-real-time tasks. IEEE Transactions on Software Engineering 15(12), 1497–1506 (1989)
- 78. Devi, U., Anderson, J.: Tardiness bounds ~~for~~under global EDF scheduling on a multiprocessor. In: Proceedings of the IEEE Real-Time Systems Symposium, pp. 330–341. IEEE Computer Society Press, Miami, FL (2005)

Page 222:

- 137. Li, J., ~~Saifullah, A.~~Chen, J. J., Agrawal, K., ~~Gill, C., Lu, C.~~Lu, C., Gill, C., Saifullah, A.: ~~Capacity augmentation bound of federated scheduling for parallel dag tasks.~~Analysis of Federated and Global Scheduling for Parallel Real-Time Tasks. In: Proceedings of the 2012 26th Euromicro Conference on Real-Time Systems, ECRTS '14. IEEE Computer Society Press, Madrid (Spain) (2014)