

# 基于遗传算法分组竞争的列表调度算法

汪 敏

合肥工业大学微电子学院, 安徽 合肥

收稿日期: 2022年5月5日; 录用日期: 2022年5月28日; 发布日期: 2022年6月8日

## 摘 要

遗传算法是有效解决任务调度问题的主流算法之一。针对传统遗传算法存在的过早收敛以及搜索空间不够等问题, 提出了一种自适应遗传算子, 去重机制以及分组竞争策略融合的改进算法GCGA, 该算法根据选取个体之间的差异程度决定个体的交叉变异的概率, 且对种群中的重复个体采用去重判断, 提高种群的多样性; 同时采用多子代竞争, 提高个体质量与搜索效率。实验表明, 与IGA算法相比, 平均性能可提升25.84%。

## 关键词

任务调度, 遗传算法, 分组竞争

# List Scheduling Algorithm Based on Genetic Algorithm Group Competition

Min Wang

School of Microelectronics, Hefei University of Technology, Hefei Anhui

Received: May 5<sup>th</sup>, 2022; accepted: May 28<sup>th</sup>, 2022; published: Jun. 8<sup>th</sup>, 2022

## Abstract

Genetic algorithm is one of the mainstream algorithms to effectively solve task scheduling problems. Aiming at the problems of premature convergence and insufficient search space in traditional genetic algorithms, an improved algorithm GCGA, which combines adaptive genetic operator, deduplication mechanism and grouping competition strategy, is proposed. The algorithm is determined according to the degree of difference between selected individuals. The probability of individual crossover variation, and deduplication judgment is used for repeated individuals in the population to improve the diversity of the population; at the same time, multi-offspring competi-

tion is used to improve individual quality and search efficiency. Experiments show that the average performance can be improved by 25.84% compared with the IGA algorithm.

## Keywords

Task Scheduling, Genetic Algorithm, Group Competition

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

作为多核技术发展的关键问题，如何开发一个高效的调度策略成为提高多核工作性能和效率的关键因素之一。

任务调度的目的是将待执行的任务合理的分配到合适的处理器上以便获得最短任务执行时间及其它性能的优化。任务调度问题属于 NP 完全问题，针对此类问题众多学者们做出了广泛的研究，分别提出了以列表调度算法(HEFT [1], TPIA [2])、任务复制调度算法(TDSA-RC [3], IDSRR [4], IREA [5])和任务簇调度算法(TDRECS [6], DCP [7])为主的启发式调度算法以及随机搜索类算法[8] [9] [10] [11]。其中，启发式调度算法存在任务优先级列表单一，无法反映调度解的整体构成情况，导致获得的为局部最优解，而以遗传算法为主的智能搜索算法能够较好的解决这一情况。

遗传算法(GA) [12]是上世纪 70 年代由 Holland 教授提出的一种模拟自然界生物进化的算法，通过模拟自然界生物的进化过程，将待求取问题的可行解抽象成种群的个体，利用进化现象繁殖更多的个体，找到最优解。因其具有简单性、通用性以及并行性等特点成为了解决任务调度的重要算法之一，但是也存在一些缺陷，例如传统的遗传算法在进化的过程采用固定的遗传概率与遗传算子，在进化的后期可能会造成对优良个体的破坏导致算法过早收敛，陷入局部最优的情况。姚虎[13]等人提出了一种改善型的遗传算法 IGA，该算法提出了三段式选择法与交叉区域的相似度和无性繁殖，虽然在进化过程在减少了优秀个体不被破坏的机率，但是进化初期由于种群质量不高，导致算法搜索时间过长；且采用固定的遗传因子无法提高个体的多样性，导致算法过早收敛。基于此，本文提出了基于遗传算法分组竞争的列表调度算法，该算法以已有的列表调度为基础，通过交换算子提高初始种群的质量，使得种群的个体在较优解的区域附近，减少搜索过程；通过删除种群的相似个体，并增加随机扰动提高种群的多样性，保留种群中部分最优个体，同时使用多组个体进化，加快算法收敛速度，提高解的质量。

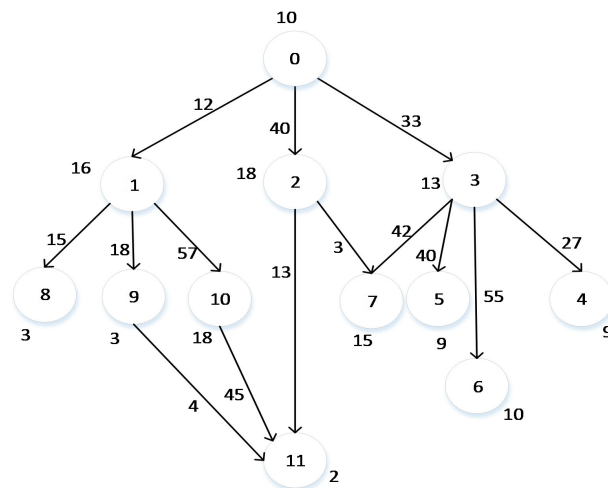
## 2. 相关模型

### 2.1. 任务模型

任务调度的目标是将各个任务合理分配到处理器内核中，使得任务执行完成所需的时间最小。对于多核系统的静态任务调度问题而言，各任务的执行时间，任务之间的通讯消耗以及任务的依赖关系是固定不变的，因此，针对此类问题可以采用有向无环图(Directed Acyclic Graph, DAG)进行建模，如图 1 所示。

同构多核系统下的任务在 DAG 图中可以采用式(1)的表达方式：

$$G = (V, E, W, C) \quad (1)$$



**Figure 1.** A DAG diagram mapped from an actual program  
**图 1.** 一种实际程序映射而来的 DAG 图

其中,  $V$  表示图中所有任务的集合,  $V = \{v_1, v_2, \dots, v_n\}$ ,  $v_i$  代表其中一个任务;  $E$  指的是任务所有边的集合,  $e(i, j) \in E$  表示任务  $i$  与任务  $j$  之间存在数据通讯关系, 即任务  $i$  是任务  $j$  的前驱, 任务  $j$  是任务  $i$  的后继;  $W$  代表的是任务的权重集合, 即每个任务在各个处理器上的执行的时间, 对于同构多核系统来说, 各任务在不同处理器上的执行时间相同, 即  $W(v_i, p_k) = W(v_i, p_l)$ ;  $C$  表示任务之间通讯时间的集合, 若两个任务在同一个处理器上则  $c(i, j) = 0$ , 否则  $c(i, j) \neq 0$ 。

## 2.2. 约束条件

在任务的调度过程中, 需要对调度的任务进行条件约束, 模拟任务在多核处理器上的执行过程, 获得真实有效的调度结果, 所以在任务调度的过程中, 任务需要满足以下约束条件:

- 1) 若两任务被分配到同一内核中, 在同一时间只能执行其中的一个任务;
- 2) 若两个任务之间存在数据通信, 只有前驱任务执行结束且两任务之间完成数据通信后, 后继任务才开始执行;
- 3) 任务的执行过程不可中断, 只有当一个任务执行结束, 才可以开始执行下一个任务。

## 3. 基于分组竞争的遗传算法

针对传统的遗传算法中存在的初始种群个体质量不高, 遗传后期搜索效率低引起的过早收敛等问题, 本文提出了一种自适应遗传算子, 去重机制以及分组竞争策略融合的改进算法 GCGA, 根据选取个体之间的差异程度决定个体的交叉变异的概率, 且对种群中的重复个体采用去重判断, 提高种群的多样性; 同时采用多子代竞争, 提高个体质量与搜索效率。

### 3.1. 算法原理及框架

遗传算法遵循的是“优胜劣汰, 适者生存”的主要思想, 这一思想主要表现在个体的选择上, 个体的适应值越高被选择的机率就越大, 其余个体被选择的机会就会大大降低, 为了提高其它个体被选择的机率, 在传统遗传算法的基础上同时选择多组个体进行变异, 增加个体被选择的机率, 提高解空间的搜索范围及搜索效率。基于此思想, 本文在传统遗传算法的基础上进行改进, 采用多组竞争的算法 GCGA 结构。在选择个体时同时从父代个体中选择两组个体进行遗传操作, 并采用两组不同的遗传方式。在进

化的初始阶段, 个体的优劣有一定的差异, 此时可以根据不同的遗传方式使得个体解朝最优解方向前进。若个体的适应值较大, 说明该个体的调度解较为优秀, 此时可以采用破坏性较小的基于单点交换的交叉算子 SPC (Single Point Cross) 和基于两点基因随机交换的变异算子 EM (Exchange Mutation), 在父代个体的较小范围内进行细致的搜索, 继承父代个体的优良基因的同时有更好的发展; 若个体的适应值较小, 说明该个体的调度解比较差, 没有进入最优解的区域范围内, 此时可以采用适合大范围搜索空间的基于随机概率的均匀交叉算子 UC (Uniform Cross) 和基于位置倒换的变异算子 RM (Reverse Mutation), 对父代基因的破坏能力更大, 从而提高获得优秀个体的机率。在种群进化的后期阶段, 采用不同遗传算子的方法同样适用。在遗传后期, 个体的适应值趋向于收敛, 若只在较小范围内搜索, 个体的基因变异的可能较小, 获得调度解趋向于局部最优解, 若同时增大个体的搜索范围, 使用具有更大破坏性的遗传算子, 从多个方向进行搜索, 更容易获得全局最优解。算法主要实现流程如图 2 所示。

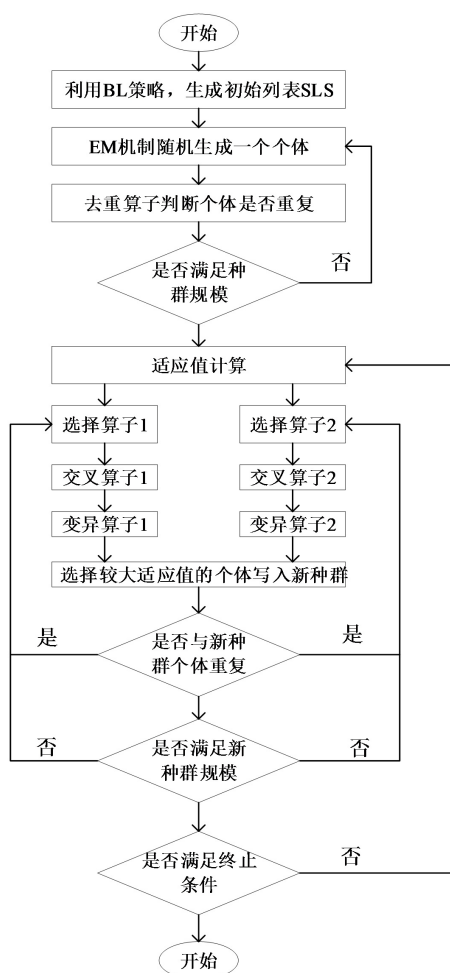


Figure 2. Framework diagram of GCGA algorithm  
图 2. GCGA 算法框架图

### 3.2. 自适应交叉率与变异率

不同交叉概率与变异概率的设置能够影响算法的整体性能。对于交叉率  $P_c$  而言, 如果数值过大, 在遗传模式下具有高适应值的个体结构可能会遭到破坏, 若取值过小会导致整个搜索过程缓慢, 难以搜索

到最优解。而变异率  $P_m$  的取值也应该选取适当, 如果取值太大, 那么遗传算法就会变成完全随机的搜索类算法, 如果太小, 个体进行变异的概率就会降低不易产生新的个体, 无法获得更佳个体。所以需要根据种群中个体适应值的改变自适应的选择交叉与变异的概率。传统的遗传算法中采用的都是固定的交叉与变异的概率值, 无法根据种群中个体的进化自适应地做出改变, 对算法性能的提升效果不明显。针对这类问题, 可以采用自适应的交叉与变异概率, 其计算方式如式(2)和式(3)所示。

$$P_c = \begin{cases} K_1 * \frac{f_{\max} - f(i)}{f_{\max}} & \text{other} \\ K_2 P(i, j) \geq 0.7 \\ K_3 P(i, j) \leq 0.3 \end{cases} \quad (2)$$

$$P_m = \begin{cases} K_4 * \frac{f_{\max} - f(i)}{f_{\max}} & \text{other} \\ K_5 P(i, j) \geq 0.7 \\ K_6 P(i, j) \leq 0.3 \end{cases} \quad (3)$$

式中  $P_c$  和  $P_m$  分别代表交叉概率与变异概率,  $f_{\max}$  为当前种群中最大的适应值,  $K_1 \sim K_6$  是一个 0 到 1 之间的常数, 其中  $K_2 > K_3$ ,  $K_5 > K_6$ , 这是因为当个体与最大适应值个体得相似率较大时, 此时个体的质量可能较差, 需要通过遗传运算获取更为优秀的子代。 $P(i, j)$  表示的是所选个体与最大适应值个体的相似率。相似率指的是两个体中基因重复的次数与任务数的比值, 计算方式如式(4)所示。依次比较两个体中相同基因位置上的值, 相同取值为 1, 不同取值为 0, 所有基因位置对应相同值得次数记为  $\Delta ij(k)$ 。

$$P(i, j) = \frac{\sum_{k=1}^n \Delta ij(k)}{NT} \quad (4)$$

$$\Delta ij(k) = \begin{cases} 0, & i(k) \neq j(k) \\ 1, & i(k) = j(k) \end{cases} \quad (5)$$

### 3.3. 去重策略

GCGA 算法意在提高解空间的搜索范围, 获取更多的任务拓扑序列, 使得任务调度的最终执行时间最短。在个体的进化过程中会获得很多的任务列表, 但是这些任务列表存在相似情况, 若种群中多个个体的相似度过高, 产生的子代个体相似度也会过高, 不利于种群的个体多样性, 减少了解空间的搜索范围。基于这一目标, GCGA 算法从提高种群中个体的多样性出发, 提出了个体相似性判断机制。对生成的子代个体与种群的个体进行相似度判断, 相似度的计算公式如(4)和(5)所示, 当  $P(i, j) = 1$  时说明个体与种群的个体重复, 不允许将此个体写入新种群, 重新选择个体进化生成新的符合要求的个体, 以便改善个体的质量, 提高搜索效率。

### 3.4. 遗传算子

GCGA 算法是一种分组竞争, 择优选取的遗传算法, 所以在个体的进化过程中采用不同的交叉与变异算子, 实现不同方向的解空间的搜索。

基于随机概率的均匀交叉算子 UC 通过概率计算对选中的个体进行大范围的搜索, 且保持良好的信息交换。首先随机生成一个个体的交换概率  $ps$ , 然后再对个体中的每一个基因生成对应的概率  $p$ , 对满足  $p > ps$  条件的基因进行对应位置基因交换。基于单点交换的交叉算子 SPC 是对某单一位置上的基因进

行交叉, 具有更小的破坏原个体的能力, 对解空间的搜索更细致。该算子通过随机选择个体的某一基因位置, 将此基因位置之后的所有基因进行交换。

基于位置倒换的变异算子 RM 是在个体基因的长度范围内, 随机选择连续位置上的部分基因, 并随机打乱这些位置上的基因。基于随机交换的变异算子 EM 是在个体基因的长度范围内, 随机选择两个位置上的基因, 并将两个位置上的基因进行交换形成新的个体, 此变异方法产生的新个体很好的继承了旧个体的大部分基因, 加速了向最优解方向前进的速度。

## 4. 实验

为了验证所提出算法的调度性能, 采用 C 实现 IGA 算法, 采用平均加速比作为评估算法性能的参数指标, 分别测试在随机任务图与实际任务图中的加速效果。平均加速比的计算方式如式(6)所示, 若加速比大于 0 时, 说明算法 A 的性能优于算法 B, 若加速比小于 0, 说明算法 B 的性能优于算法 A。

$$ACC(A, B) = \frac{\overline{SL_B} - \overline{SL_A}}{\overline{SL_A}} * 100\% \quad (6)$$

### 4.1. 随机任务图实验

根据 Kun [14]工作中生成随机任务图的流程, 本实验选取任务图中任务节点的总数(NT)、随机任务 DAG 的最大层级数(LEVEL)和通讯计算比(CCR)、处理单元的总数(NP)作为对比随机任务图时的控制参数。其中, NT 的取值为 25, 50, 75, 100 (默认值 50); LEVEL 的取值是 3, 5, 7, 9 (默认值 5); NP 的取值为 2, 3, 4, 5; CCR 的取值是 0.1, 1.0, 5.0, 10.0 (默认值 1.0)。通过控制变量法, 每组包括 100 张任务图, 总计生成 6400 张任务图。

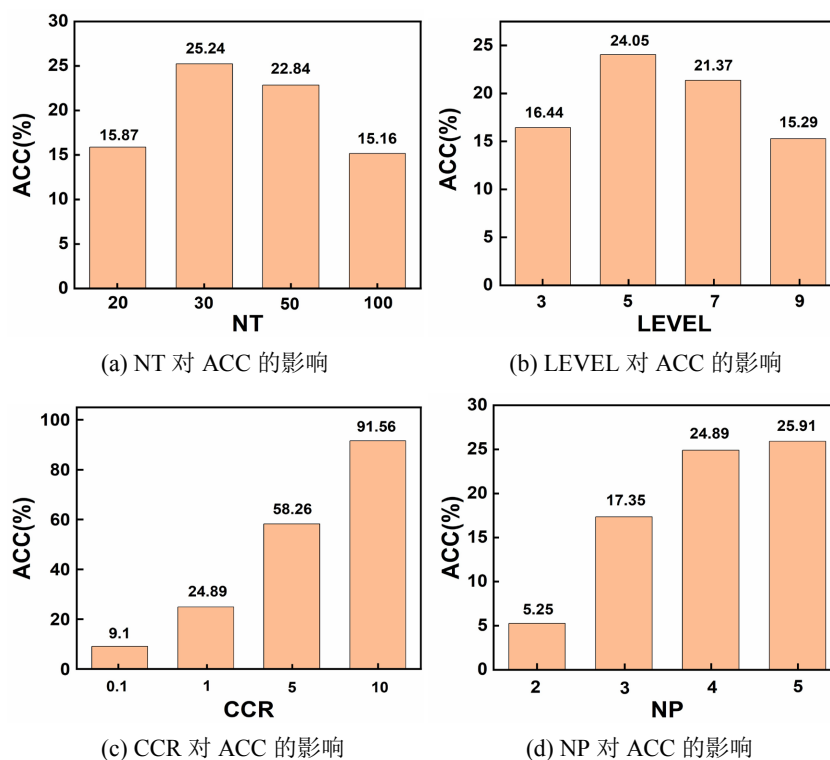
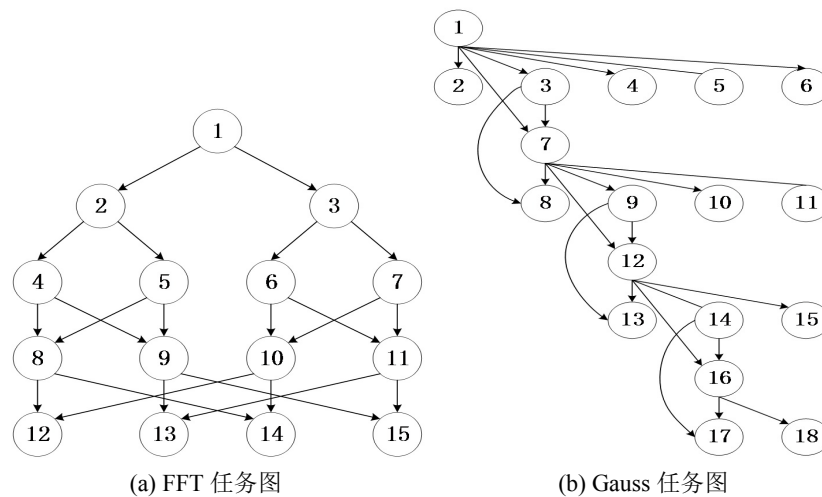


Figure 3. Random task graph experimental results  
图 3. 随机任务图实验结果

由图 3(a)~(d)的实验结果可知,在各种参数的设置范围内,无论参数 NT, LEVEL, CCR 和 NP 如何变化, GCGA 算法都能取到一个很好的调度效果,平均加速比为 25.84%。图 3(a)与图 3(b)所示,当 NT = 30 和 LEVEL = 5 时,算法的加速效果最好,而图 3(c)与图 3(d)所示,ACC 的数值与通信比 CCR 和处理器的个数 NP 成正相关,CCR 和 NP 的数值越大,调度效果越明显,且当 CCR = 10 时,任务的加速比达到了 91.56%,说明 GCGA 算法在通讯权值较大的情况下调度结果出现质的飞跃。因此,与 IGA 算法相比, GCGA 算法更适合处理通讯占比更大,任务数不多且层级数较少的任务图。

#### 4.2. 实际任务图实验

此实验是为了验证 GCGA 算法相比于 IGA 算法在处理实际应用任务图的加速效果,实验采用常见的两种任务[15]:快速傅里叶变换(FFT)和高斯消元(Gauss)和具有相同任务数的随机任务图作为对比对象,实际任务图如图 4(a)、图 4(b)所示。



值下 GCGA 算法的调度结果均优于 IGA 算法。其中, CCR 对算法的影响更明显, 任务间的通讯消耗越大, 算法的加速比越大, 调度效果更好。对于 Gauss 任务图来说, NP 的值与 ACC 成负相关, NP 值越大, GCGA 算法对 Gauss 的调度效果相对减小。对 FFT 任务图和 Random 任务图来说, 不同的 NP 取值下都能取得较好的调度效果。

## 5. 结语

针对 IGA 算法种群质量不高、算法过早收敛陷入局部最优等问题, 提出了一种基于遗传算法分组竞争的调度算法。采用 BL 策略与随机搜索的方式提高初始种群的质量, 然后采用多组个体进行相互竞争的遗传模式, 提高搜索效率, 同时判断种群个体的相似性, 删除重复个体, 提高种群的多样性, 防止种群过早收敛。从对比实验中可知, 与 IGA 算法相比, 本文采用的 GCGA 算法在随机任务图实验与实际任务图实验中都取得了较为优秀的调度结果, 平均性能可提升 25.84%, 更适合中等规模且层级较少、通讯消耗较大的任务图。

## 参考文献

- [1] Topcuoglu, H., Hariri, S. and Wu, M.Y. (2002) Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing. *IEEE Transactions on Parallel and Distributed Systems*, **13**, 260-274. <https://doi.org/10.1109/71.993206>
- [2] 张多利, 廖金月, 罗乐, 等. 一种多核系统任务扰动迭代算法[J]. 电子测量与仪器学报, 2020, 34(9): 133-139.
- [3] 王月恒. 基于任务复制的多核调度算法研究[D]: [硕士学位论文]. 合肥: 合肥工业大学, 2021.
- [4] 周超群, 周亦敏. 一种改进的基于复制的异构多核任务调度算法[J]. 电子科技, 2017, 30(6): 57-62.
- [5] 谢志强, 韩英杰, 齐永红, 等. 基于关键路径和任务复制的多核调度算法[J]. 国防科技大学学报, 2014(1): 172-177.
- [6] 任良育, 赵成萍, 严华. 基于任务复制与冗余消除的多核调度算法[J]. 计算机工程, 2019, 45(5): 59-65.
- [7] Kwok, Y.K. and Ahmad, I. (1996) Dynamic Critical-Path Scheduling: An Effective Technique for Allocating Task Graphs to Multiprocessors. *IEEE Transactions on Parallel Distributed Systems*, **7**, 506-521. <https://doi.org/10.1109/71.503776>
- [8] 王昱, 左利云. 云计算中基于改进遗传算法的多目标优化调度[J]. 广东石油化工学院学报, 2020, 30(1): 5.
- [9] 白楠, 陈正阳. 基于改进模拟退火算法的片上网络任务调度优化研究[J]. 河南科技, 2018(25): 2.
- [10] 姚英彪, 王璇. 面向多核任务调度的混合遗传算法[J]. 系统工程与电子技术, 2015(8): 1928-1935.
- [11] 张皓, 赵开新, 孙冬. 改进遗传算法在云资源任务调度中的应用[J]. 河南工学院学报, 2020, 28(3): 24-29. <https://doi.org/10.26549/jxjfxysj.v3i7.4747>
- [12] Hou, E. and Ansari, N. (1994) Genetic Algorithm for Multiprocessor Scheduling. *IEEE Transactions on Parallel and Distributed Systems*, **5**, 113-120. <https://doi.org/10.1109/71.265940>
- [13] 姚虎. 云计算环境下基于改进遗传算法的任务调度策略研究[Z]. 内蒙古农业大学, 2018.
- [14] He, K., Meng, X., Pan, Z., et al. (2019) A Novel Task-Duplication Based Clustering Algorithm for Heterogeneous Computing Environments. *IEEE Transactions on Parallel Distributed Systems*, **30**, 2-14. <https://doi.org/10.1109/TPDS.2018.2851221>
- [15] Amoura, A.K., Bampis, E. and Konig, J. (1998) Scheduling Algorithms for Parallel Gaussian Elimination with Communication Costs. *IEEE Transactions on Parallel and Distributed Systems*, **9**, 679-686. <https://doi.org/10.1109/71.707547>