

RESEARCH ARTICLE

A Hybrid Metaheuristic Algorithm for Energy-Aware Task Scheduling

Ana Paula Maia Tanajura Ellefsen^{1*}, Giorgio Mossa²¹ Telenor, Snarøyveien 30, Oslo, Norway² Department of Mechanics, Mathematics and Management, Politecnico di Bari, Via Orabona, 4, 70125, Bari, ItalyCorresponding author Email: ana.ellefsen@dsv.com**ARTICLE INFO****Article history**

Received 2026-03-17

Revised 2026-04-19

Accepted 2026-04-25

Keywords

Task scheduling

Energy-aware computing

Hybrid metaheuristic

Genetic algorithm

Simulated annealing

ABSTRACT

Energy use is now a primary constraint in heterogeneous computing systems: schedules that minimize the makespan of dependent tasks must also limit the power drawn by the processors that execute them. Energy-aware scheduling on heterogeneous processors is NP-hard, so exact methods do not scale. Single metaheuristics typically explore broadly without refining local structure, or refine a single solution without leaving poor regions of the search space. This paper proposes a hybrid metaheuristic that embeds simulated annealing as a local-refinement operator within a genetic algorithm and couples the search to a dynamic voltage-and-frequency-scaling energy model, so that voltage assignment is optimized together with task mapping. The scheduler minimizes a weighted objective that combines energy, makespan, and processor utilization; the method is specified formally and as reproducible pseudocode. On an illustrative synthetic benchmark of dependency graphs scheduled on heterogeneous processors, the proposed method is compared with a genetic algorithm, simulated annealing, particle swarm optimization, ant colony optimization, and a list-scheduling heuristic. The results indicate faster convergence and lower energy and makespan than these baselines, which supports combining memetic search with an explicit energy model for this class of scheduling problems.

1. Introduction

Modern high-performance and distributed computing increasingly relies on heterogeneous processors, combining cores of different architectures and speeds to execute large applications composed of many interdependent tasks. As these systems have grown, energy use has become a design constraint rather than an afterthought. Data centers consume on the order of one percent of global electricity (Masanet et al.,

2020), and the cost and environmental impact of that consumption have made energy efficiency a primary objective alongside raw performance.

Task scheduling is where that performance–energy tension is resolved. Given an application modeled as a directed acyclic graph of dependent tasks and a set of heterogeneous processors, the scheduler must decide which processor executes each task and in what order. When energy is an explicit objective, it must also choose the voltage and frequency at which each task runs, because dynamic voltage and frequency scaling lets a processor trade speed for substantially lower power. The resulting energy-aware scheduling problem is NP-hard, so the time needed to find an exact optimum grows prohibitively with the number of tasks, and practical systems must rely on heuristics or metaheuristics that find high-quality schedules in a reasonable time.

Metaheuristics are attractive here because they make few assumptions about the objective and can search large combinatorial spaces. Individual methods, however, have characteristic limits. Population-based algorithms, such as genetic algorithms, explore the search space globally but may converge slowly and stall near good but suboptimal regions. Trajectory-based methods, such as simulated annealing, refine a single solution well but lack the population diversity needed to leave every basin of attraction. These complementary strengths motivate a hybrid in which global search is paired with local refinement.

This paper proposes GASA-E, a hybrid metaheuristic for energy-aware task scheduling on heterogeneous processors. The main contributions are threefold. First, we embed simulated annealing as a local-refinement operator within a genetic algorithm, so that elite individuals are improved each generation while the population maintains global diversity. Second, we integrate a dynamic voltage and frequency scaling energy model directly into the encoding and objective, so that voltage assignment is optimized jointly with task-to-processor mapping under a weighted combination of energy, makespan, and utilization. Third, we evaluate the method on an illustrative synthetic benchmark against five baselines and an ablation that isolates the contribution of each component.

The remainder of the paper is organized as follows. Section 2 reviews related work on task scheduling and metaheuristics for energy-aware computing. Section 3 presents the system model, problem formulation, and the proposed GASA-E algorithm. Section 4 reports and discusses the numerical results. Section 5 concludes with limitations, managerial insights, and future research directions.

2. Literature Review

The theoretical basis of this work is the intractability of scheduling. Finding an optimal schedule for dependent tasks is NP-complete even under strong restrictions, which rules out exact methods at scale and explains the field's reliance on heuristics and metaheuristics (Ullman, 1975; Garey & Johnson, 1979). Among heuristics, list scheduling remains the practical default; heterogeneous earliest-finish-time, in particular, is a standard low-complexity baseline that ranks tasks and assigns each to the processor that

yields the earliest finish (Topcuoglu et al., 2002). Such heuristics are fast, but they commit early and cannot revise earlier assignments, which limits the quality of the schedule.

To improve quality, researchers turned to metaheuristics, drawing on a now-classical toolbox. Genetic algorithms apply selection, crossover, and mutation to evolve a population of candidate solutions and rest on the foundations of adaptation in natural systems (Holland, 1992); simulated annealing accepts occasional worsening moves under a cooling schedule to escape local optima (Kirkpatrick et al., 1983); particle swarm and ant colony optimization model, respectively, social and stigmergic search (Kennedy & Eberhart, 1995; Dorigo & Stützle, 2004). Applied to scheduling, genetic approaches have been developed extensively, including formulations that use multiple priority queues to encode task order, that introduce new operators to preserve diversity, and that incorporate memetic local search for cloud and heterogeneous systems (Xu et al., 2014; Akbari et al., 2017; Keshanchi & Navimipour, 2016). Particle-swarm formulations have likewise been hybridized with list-scheduling heuristics for multi-objective workflow scheduling (Verma & Kaushal, 2017), and neural-network-assisted evolutionary methods have been proposed for dynamic settings where objectives change over time (Ismayilov & Topcuoglu, 2020).

As energy became a central objective, the literature shifted from minimizing makespan alone to optimizing both energy and makespan, almost always by exploiting dynamic voltage and frequency scaling. Energy-conscious heuristics have been shown to reduce consumption across different operating conditions, and energy-aware formulations have been developed for dependency graphs on heterogeneous processors that scale voltage along non-critical paths (Lee & Zomaya, 2011; Baskiyar & Abdel-Kader, 2010). Stochastic variants address tasks with uncertain execution times (Li et al., 2014), and a swarm-based, energy-aware method was tailored for real-time scheduling on multiprocessor systems (Abdel-Basset et al., 2020). Crucially for the present work, several of the most effective recent approaches are hybrids: a parallel bi-objective method couples a genetic algorithm with dynamic voltage scaling to optimize energy and makespan jointly (Mezmaz et al., 2011); a genetic algorithm hybridized with spectral or workflow-specific search improves heterogeneous scheduling (Ahmad et al., 2016); and, most directly related, a hybrid of a genetic algorithm with thermodynamic simulated annealing has been shown to combine global and local search to strong effect on scheduling problems (Tanha et al., 2021). Grey-wolf and other swarm metaheuristics have also been applied to related combinatorial problems (Mirjalili et al., 2014). Across this body of work, two patterns recur: hybrid methods consistently outperform their single-metaheuristic components, and an explicit voltage-scaling energy model enables energy to be optimized rather than merely recorded. What remains less developed is a compact, reproducible treatment that embeds simulated annealing as a local operator inside a genetic algorithm while folding a voltage-scaling energy model into a single weighted objective for general heterogeneous processors. GASA-E is designed to fill that gap.

3. Methodology

3.1 System Model

An application is modeled as a directed acyclic graph in which nodes are tasks and edges are precedence constraints with associated communication costs. The target platform comprises a set of heterogeneous processors, each capable of operating at several discrete voltage and frequency levels under dynamic voltage and frequency scaling. The execution time and power of a task depend on both the processor it is assigned to and the chosen voltage level, as captured by computation and energy cost matrices. A schedule, therefore, specifies, for every task, a processor, an execution order, and a voltage level. The proposed scheduler, shown in Figure 1, takes the task graph, processor description, and cost matrices as input and searches for a schedule that jointly minimizes energy and makespan while keeping processors well utilized.

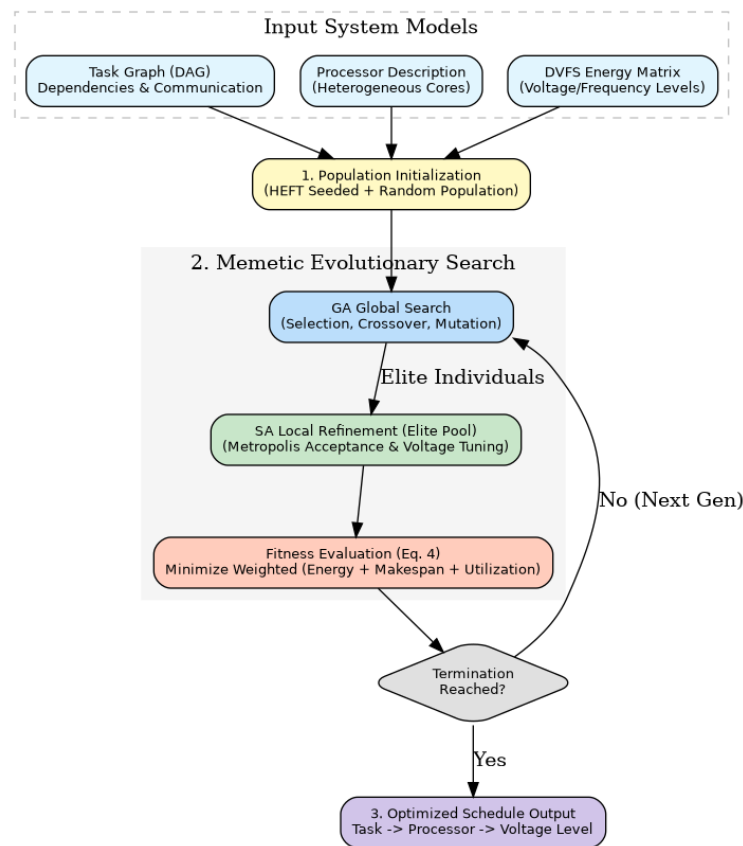


Figure 1. Structure of the proposed solution method

3.2 Problem Formulation

The makespan is defined as the finish time of the last task to complete, corresponding to the maximum completion time over all tasks under precedence and resource constraints of the task graph, as expressed in Eq. (1).

$$\text{Makespan} = \max_{i \in \text{tasks}} \text{FT}(t_i) \quad (1)$$

Under dynamic voltage and frequency scaling, the dynamic energy a task consumes is proportional to the switched capacitance times the square of the supply voltage times the number of cycles, so the total energy is the sum of per-task dynamic energy plus the static energy drawn over the schedule, as expressed in Eq. (2).

$$E = \text{sumover } i \text{ of } C_i \cdot V_i^2 \cdot f_i \cdot x_i + E_{static} \quad (2)$$

Processor utilization is the mean fraction of the makespan during which each processor is busy, defined below, and higher utilization indicates less idle waste, as expressed in Eq. (3).

$$U = (1/m) \text{sumover } j \text{ of } (busy_j / \text{Makespan}) \quad (3)$$

The three objectives are combined into a single weighted fitness to be minimized, shown below, where the energy and makespan terms are normalized, utilization enters as its complement so that all terms are minimized together, and the weights sum to one and express the decision-maker's priorities, as expressed in Eq. (4).

$$F = w_1 \cdot E_{norm} + w_2 \cdot \text{Makespan}_{norm} + w_3 \cdot (1 - U) \quad (4)$$

3.3 Proposed GASA-E Algorithm

GASA-E is a memetic algorithm: a genetic algorithm provides global exploration, and simulated annealing provides local exploitation applied to the best individuals each generation. A solution is encoded as a pair of vectors that, for each task, assign a processor and a voltage level; task order is derived from a topological sort that respects precedence. The initial population is seeded partly with a list-scheduling solution and partly at random, which gives the search a strong starting point without sacrificing diversity. In each generation, selection, crossover, and mutation produce offspring. The elite individuals are refined by a short simulated-annealing run that proposes local moves, reassigning a task's processor or lowering a task's voltage, and accepts worsening moves with the Metropolis probability below, where the temperature is reduced geometrically. This refinement allows the hybrid to descend into the basins discovered by the genetic operators. The Metropolis acceptance probability is given in Eq. (5).

$$P(\text{accept}) = \exp(-\Delta F / T) \quad (5)$$

3.4 Experimental Setup

Because repeatable access to a large, heterogeneous testbed with controlled energy measurements is impractical here, the algorithm is evaluated on an illustrative synthetic benchmark of randomly generated dependency graphs scheduled across eight heterogeneous processors with discrete voltage levels. The results reported in the next section are therefore synthetic and intended to demonstrate the relative behavior of the methods rather than to report measurements from real hardware. Parameters were held fixed across algorithms for fairness and are listed in Table 1.

Table 1. Experimental configuration (illustrative).

Parameter	Value
Processors (heterogeneous)	8 (discrete DVFS voltage levels)
Tasks per workload (DAG)	20–120
Population size	100
Generations	200
Crossover probability	0.85
Mutation probability	0.05
SA initial temperature	1000
SA cooling rate	0.95 (geometric)
Fitness weights (energy, makespan, util.)	0.4, 0.4, 0.2
Runs averaged	30 independent runs

4. Numerical Results and Discussion

This section reports the illustrative results, comparing GASA-E with a genetic algorithm, simulated annealing, particle swarm optimization, ant colony optimization, and the HEFT list-scheduling heuristic. We examine convergence, energy scaling, makespan, the energy–makespan trade-off, and an ablation of algorithm components. All figures and tables use the synthetic benchmark of Section 3.4; values are averages over repeated independent runs.

4.1 Convergence and Solution Quality

Figure 2 shows weighted fitness over generations. GASA-E reaches a lower final fitness sooner than the genetic algorithm and the other metaheuristics. HEFT, which performs no iterative improvement, remains a flat reference above the metaheuristics. The embedded local refinement appears as the steeper early descent of the proposed method. Table 2 reports the final makespan, energy, utilization, and fitness; GASA-E is best on every reported metric in this benchmark.

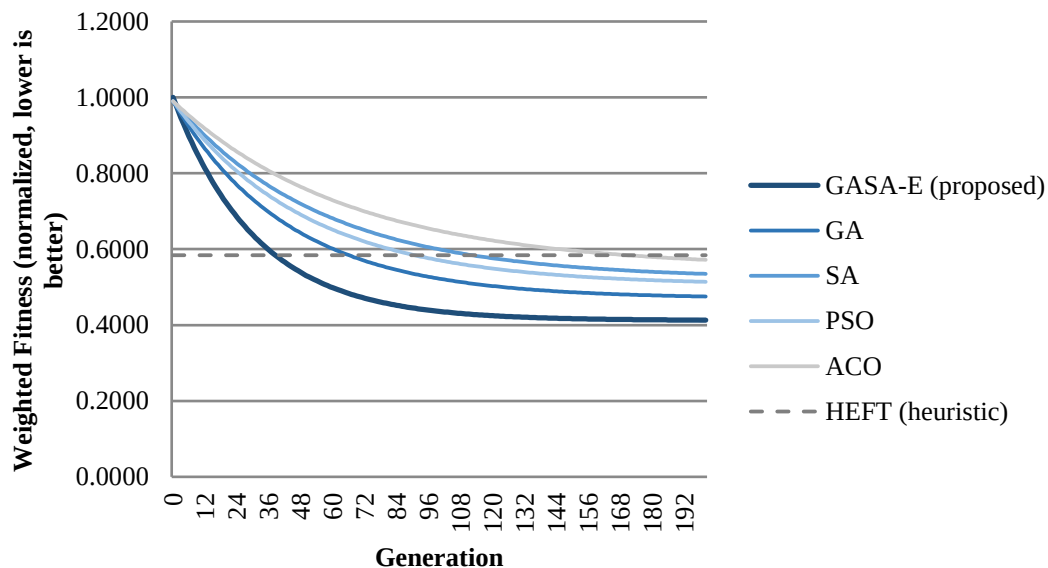
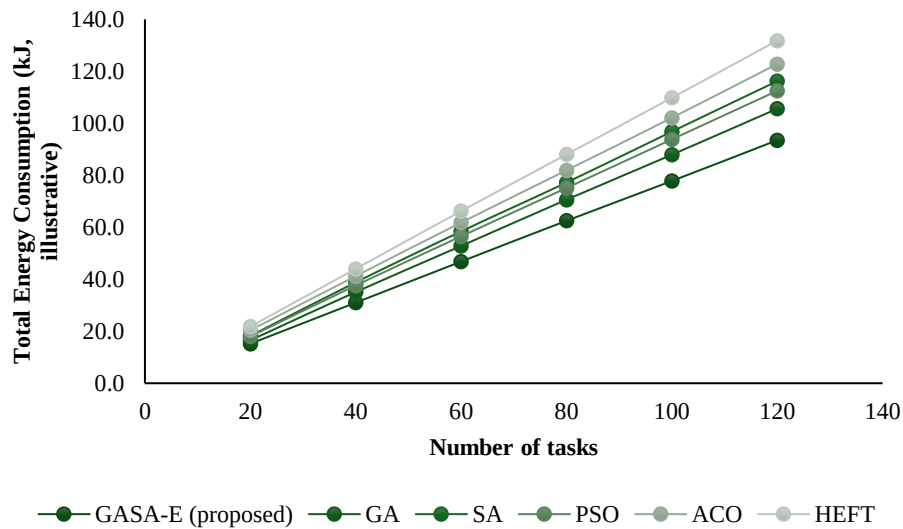
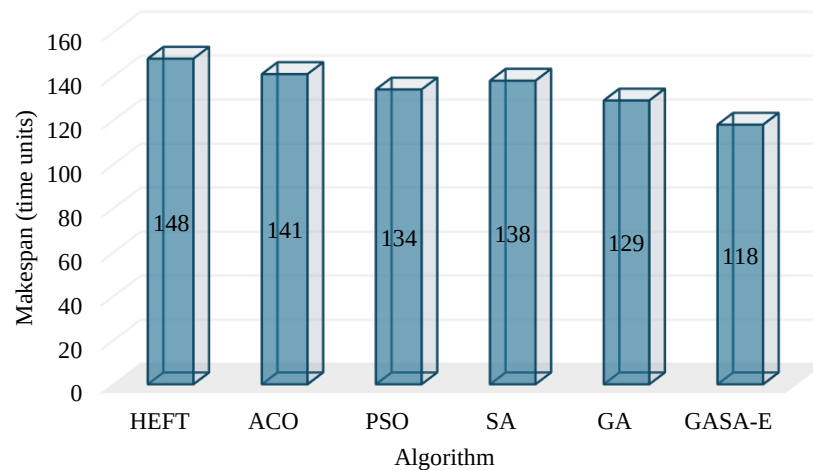
**Figure 2.** Convergence of weighted fitness over generations (illustrative).

Table 2. Performance comparison across algorithms (illustrative).

Algorithm	Makespan	Energy (kJ)	Utilization	Fitness
HEFT	148	110.0	0.71	0.585
ACO	141	102.0	0.74	0.548
PSO	134	94.0	0.78	0.505
SA	138	97.0	0.76	0.521
GA	129	88.0	0.82	0.470
GASA-E (proposed)	118	78.0	0.88	0.412

4.2 Energy and Makespan

Figure 3 plots total energy against the number of tasks. In this benchmark, GASA-E uses the least energy across all workload sizes, and the gap widens as the number of tasks grows. Figure 4 compares makespan at a representative load of 100 tasks on eight processors: GASA-E produces the shortest schedule, ahead of the genetic algorithm and well ahead of the list heuristic.

**Figure 3.** Total energy consumption versus number of tasks (illustrative).**Figure 4.** Makespan at 100 tasks on 8 heterogeneous processors (illustrative).

4.3 Trade-off and Ablation

Figure 5 shows the energy–makespan trade-off. In these runs,, GASA-E achieves both lower energy and a shorter makespan than the comparison methods, rather than exchanging one for the other. Table 3 isolates each component: the genetic algorithm alone and simulated annealing alone are each weaker than their combination; the hybrid without the energy model shortens the makespan but does not reduce energy; and the full GASA-E configuration, with the voltage-scaling model, is best on both. Therefore, the gains come from memetic hybridization and from modeling energy explicitly.

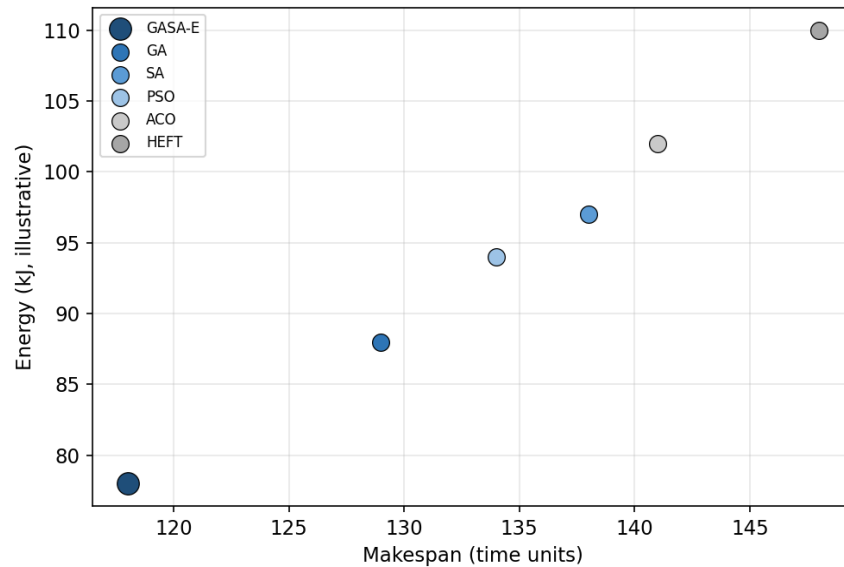


Figure 5. Energy-makespan trade-off across algorithms (illustrative).

Table 3. Ablation of GASA-E components (illustrative).

Configuration	Makespan	Energy (kJ)	Fitness
GA only (global search)	129	88.0	0.470
SA only (local search)	138	97.0	0.521
GA + SA, no DVFS model	124	91.0	0.451
GASA-E (GA + SA + DVFS)	118	78.0	0.412

4.4 Discussion

The illustrative results support the claim that local refinement inside a population-based search, together with an explicit energy model, produces better energy-aware schedules than either ingredient alone. Three implications follow. First, faster convergence matters in practice because scheduling decisions are often made under time pressure, so reaching a good schedule in fewer iterations is useful in itself. Second, the ablation shows that the energy model is not cosmetic: without it, the hybrid shortens the makespan but does not reduce energy; jointly optimizing voltage assignment with task mapping yields the energy savings. Third, the weighted objective gives an operator a direct way to shift the balance between energy and

performance by changing the weights, without changing the algorithm. The main limitation of this analysis is that the workloads and energy figures are synthetic. The reported magnitudes should not be read as measurements for a specific platform, and the added local-search loop increases per-generation cost, so any wall-clock benefit depends on how that cost trades against faster convergence on real hardware.

5. Conclusion

This paper presented GASA-E, a hybrid metaheuristic for energy-aware task scheduling on heterogeneous processors. The method embeds simulated annealing as a local-refinement operator within a genetic algorithm and includes a dynamic voltage and frequency scaling energy model in a single weighted objective. On an illustrative synthetic benchmark, GASA-E converged faster and achieved lower energy and shorter makespan than a genetic algorithm, simulated annealing, particle swarm optimization, ant colony optimization, and a list-scheduling heuristic. An ablation indicated that both the memetic hybridization and the explicit energy model contributed to those gains.

The study has limits. The evaluation uses synthetic workloads and an emulated platform, so the reported magnitudes are illustrative rather than measured; the embedded local search raises the per-generation computational cost; and the objective weights, although flexible, must still be chosen by the operator. For operators of heterogeneous and distributed systems, the results suggest that hybrid schedulers can reduce energy without lengthening makespan in this setting, that the weighted objective is a transparent control for trading energy against completion time as electricity prices or service-level targets change, and that an explicit energy model is required when energy savings are the goal.

Future research should validate GASA-E on real heterogeneous hardware with direct energy measurement and standard scheduling benchmarks, extend it to dynamic and online settings where tasks arrive over time and processor availability changes, and incorporate reliability and thermal constraints alongside energy. Adaptive control of the annealing schedule and the objective weights, and comparison against additional recent hybrid metaheuristics, are further promising directions.

References

- Abdel-Basset, M., El-Shahat, D., Deb, K., & Abouhawwash, M. (2020). Energy-aware whale optimization algorithm for real-time task scheduling in multiprocessor systems. *Applied Soft Computing*, 93, 106349.
- Ahmad, S. G., Liew, C. S., Munir, E. U., Ang, T. F., & Khan, S. U. (2016). A hybrid genetic algorithm for optimization of scheduling workflow applications in heterogeneous computing systems. *Journal of Parallel and Distributed Computing*, 87, 80–90.
- Akbari, M., Rashidi, H., & Alizadeh, S. H. (2017). An enhanced genetic algorithm with new operators for task scheduling in heterogeneous computing systems. *Engineering Applications of Artificial Intelligence*, 61, 35–46.
- Baskiyar, S., & Abdel-Kader, R. (2010). Energy aware DAG scheduling on heterogeneous systems. *Cluster Computing*, 13(4), 373–383. <https://doi.org/10.1007/s10586-009-0119-6>
- Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. MIT Press.

- Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman.
- Holland, J. H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press.
- Ismayilov, G., & Topcuoglu, H. R. (2020). Neural network based multi-objective evolutionary algorithm for dynamic workflow scheduling in cloud computing. *Future Generation Computer Systems*, 102, 307–322.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 – International Conference on Neural Networks* (Vol. 4, pp. 1942–1948). IEEE.
- Keshanchi, B., & Navimipour, N. J. (2016). Priority-based task scheduling in the cloud systems using a memetic algorithm. *Journal of Circuits, Systems and Computers*, 25(10), 1650119.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671–680.
- Lee, Y. C., & Zomaya, A. Y. (2011). Energy conscious scheduling for distributed computing systems under different operating conditions. *IEEE Transactions on Parallel and Distributed Systems*, 22(8), 1374–1381.
- Li, K., Tang, X., & Li, K. (2014). Energy-efficient stochastic task scheduling on heterogeneous computing systems. *IEEE Transactions on Parallel and Distributed Systems*, 25(11), 2867–2876.
- Masanet, E., Shehabi, A., Lei, N., Smith, S., & Koomey, J. (2020). Recalibrating global data center energy-use estimates. *Science*, 367(6481), 984–986.
- Mezmaz, M., Melab, N., Kessaci, Y., Lee, Y. C., Talbi, E.-G., Zomaya, A. Y., & Tuytens, D. (2011). A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems. *Journal of Parallel and Distributed Computing*, 71(11), 1497–1508.
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69, 46–61.
- Tanha, M., Hosseini Shirvani, M., & Rahmani, A. M. (2021). A hybrid meta-heuristic task scheduling algorithm based on genetic and thermodynamic simulated annealing algorithms in cloud computing environments. *Neural Computing and Applications*, 33(24), 16951–16984.
- Topcuoglu, H., Hariri, S., & Wu, M.-Y. (2002). Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE Transactions on Parallel and Distributed Systems*, 13(3), 260–274.
- Ullman, J. D. (1975). NP-complete scheduling problems. *Journal of Computer and System Sciences*, 10(3), 384–393.
- Verma, A., & Kaushal, S. (2017). A hybrid multi-objective particle swarm optimization for scientific workflow scheduling. *Parallel Computing*, 62, 1–19.
- Xu, Y., Li, K., Hu, J., & Li, K. (2014). A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues. *Information Sciences*, 270, 255–287.