Adaptive Genetic Algorithm with Optimized Operators for Scheduling in Computer Systems

Yulia Zakharova, Maria Sakhno

Sobolev Institute of Mathematics SB RAS, Omsk, Russia
The research was supported by Russian Science Foundation grant N 22-71-10015, https://rscf.ru/en/project/22-71-10015/.

Report Structure

- Genetic Algorithm with Generational Scheme
- Crossover and Mutation Operators
- Problem Statements
- Results for Genetic Algorithm
- Results for Genetic Algorithm with Adaptation
- Results for Genetic Algorithm with Optimized Crossover
- Conclusions and Further Research

Genetic Algorithm with Generational Scheme

1: Construct the initial population $P^0 = \{\pi_j^0\}$ of k permutations. Save n_e individuals with the best objective values as elites of P^0. Put $t = 0$.

2: Until termination condition is met, perform

2.1 for $i \leftarrow 1$ to $(k - n_e)/2$

2.1.1 Select two parent permutations π^1 and π^2 using operator $Sel(P^t)$.

2.1.2 Construct $(\pi^{1\prime}, \pi^{2\prime}) = Cross(\pi^1, \pi^2)$.

2.1.3 Apply the mutation operator to constructed permutations: $Mut(\pi^{1\prime})$ and $Mut(\pi^{2\prime})$ and save the result as individuals $\pi_{2i-1}^{t+1}, \pi_{2i}^{t+1}$ for population P^{t+1}.

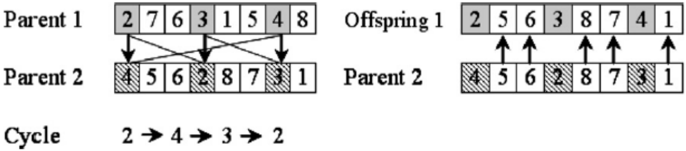
2.2 Copy elites of P^t to P^{t+1} and identify elites of P^{t+1}.

2.3 Put $t = t + 1$.

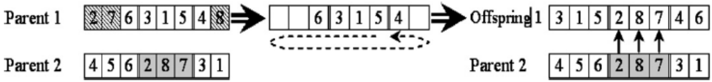
3: Return the best found individual.

Crossover Operators

Cycle Crossover (CX)

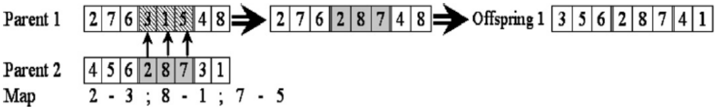


Order Crossover (OX)

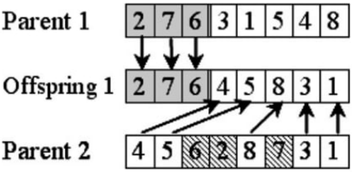


Crossover Operators

Partially Mapped Crossover (PMX)

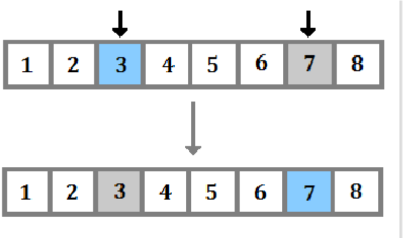


One Point Crossover (1PX)

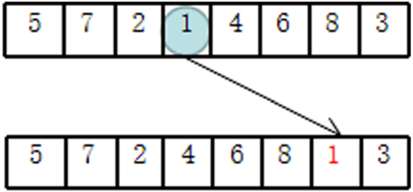


Mutation Operators

Exchange (swap) mutation



Shift (insert) mutation



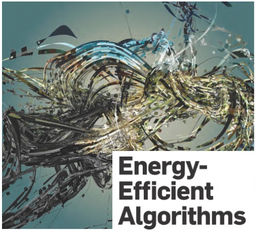
Speed Scaling Scheduling

Processors and Jobs

m is the number of speed-scalable processors
$\mathcal{J} = \{1, \ldots, n\}$ is the set of jobs:
V_j is the processing volume (work) of job j
$size_j$ is the number of processors required by job j
$W_j := \frac{V_j}{size_j}$ is the work on one processor

Parameters

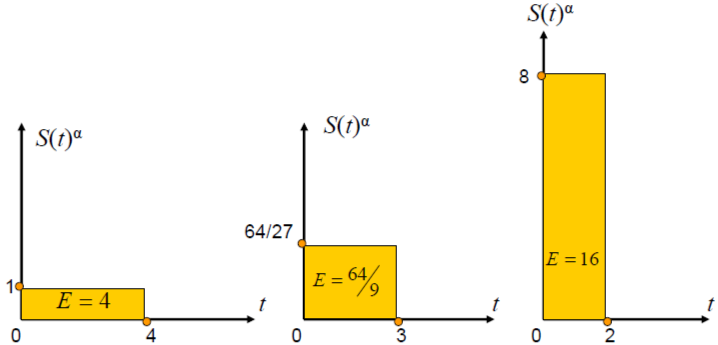
Preemption and migration are characterized for the systems with single image of the memory.
Non-preemptive instances arise in systems with distributed memory.



Homogeneous Model in Speed-scaling

If a processor runs at speed s then the energy consumption is s^α units of energy per time unit, where $\alpha > 1$ is a constant (practical studies show that $\alpha \leq 3$).

It is supposed that a continuous spectrum of processor speeds is available.

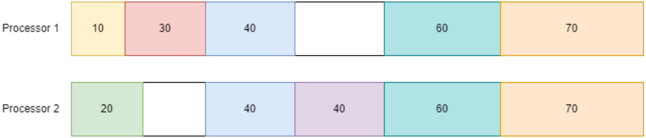


Scheduling Problem

$m = 2$, E is the energy budget.

The aim is to find a feasible schedule with the minimum total completion time so that the energy consumption is not greater than a given energy budget.

Solution



Lower Bound

Previous Research

Scheduling

- Kononov & Zakharova: Speed scaling scheduling of multiprocessor jobs with energy constraint and total completion time criterion (2023)
- Lee & Cai Scheduling one and two-processor tasks on two parallel processors (1999)
- Zakharova & Sakhno: Heuristics with local improvements for two-processor scheduling problem with energy constraint and parallelization (2024)

Evolutionary Computation

- Eremeev & Kovalenko: A memetic algorithm with optimal recombination for the asymmetric travelling salesman problem (2020)
- Neri & Cotta: Memetic Algorithms and Memetic Computing Optimization: A Literature Review (2012)
- Blum & Eremeev & Zakharova: Hybridizations of evolutionary algorithms with Large Neighborhood Search (2022)

Results of Genetic Algorithm without Adaptation

30 instances, $n = 50$

Parameter values of genetic algorithm

Parameter name	Parameter value
k	200
n_e	2
P_{IPRand}	0.2
$Selection$	Ranking
P_{Cross}	0.8
$Crossover$	1PX
P_{Mut}	0.2
$Mutation$	Shift (insert)

Relative deviation of objective function found by the GA from the lower bound

avg: 2.03%
min: 0.83%
max: 3.83%

Genetic Algorithm with Adaptation

1: Construct the initial population $P^0 = \{\pi_j^0\}$ of k permutations. Save n_e individuals with the best objective values as elites of P^0. Put $t = 0$.

2: Until termination condition is met, perform

2.1 for $i \leftarrow 1$ to $(k - n_e)/2$

2.1.1 Select two parent permutations π^1 and π^2 using operator $Sel(P^t)$.

2.1.2 Choose crossover operator and construct $(\pi^{1\prime}, \pi^{2\prime}) = Cross(\pi^1, \pi^2)$.

2.1.3 Update the weight of the chosen crossover.

2.1.4 Apply the mutation operator to constructed permutations: $Mut(\pi^{1\prime})$ and $Mut(\pi^{2\prime})$ and save the result as individuals $\pi_{2i-1}^{t+1}, \pi_{2i}^{t+1}$ for population P^{t+1}.

2.2 Copy elites of P^t to P^{t+1} and identify elites of P^{t+1}.

2.3 Put $t = t + 1$.

3: Return the best found individual.

Adaptive Technique

1: Choose a crossover. The probability of choosing each operator is proportional to its weight.
2: Apply chosen crossover to the parent genotypes.
3: Update the weight of the chosen crossover:

$$\phi_a = \begin{cases} w_1, \text{if the new solution is a new global best,} \\ w_2, \text{if the new solution is better than the current one,} \\ w_3, \text{if the new solution is better than one of the parents or both.} \end{cases}$$

$$\rho_a = \lambda \rho_a + (1 - \lambda)\phi_a.$$

Results of Genetic Algorithm with Adaptation

30 instances, $n = 50$

Relative deviation of objective function found by the GA with Adaptation from the lower bound

Leading Crossover Operator: 1PX

avg: 2.06%
min: 0.83%
max: 3.88%

GA without adaptation

avg: 2.03%
min: 0.83%
max: 3.83%

Optimal Recombination Problem (ORP)

Given two parent solutions p^1 and p^2. It is required to find a solution p' such that:

(I) $p'_i = p^1_i$ or $p'_i = p^2_i$ for all $i = 1, \ldots, k$;

(II) p' has the minimum value of objective function $\sum C_j(p)$ among all solutions that satisfy condition (I).

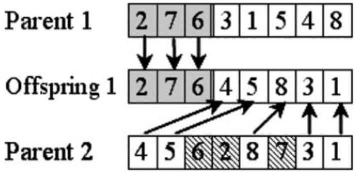
Optimal recombination may be considered as a best-improving move in a large neighbourhood defined by two parent solutions.

Property

Partial order given by the permutation.

Optimized Crossovers

One Point Crossover (1PX)



Results of Genetic Algorithm with Optimized Crossovers

GA with optimized versions of 1PX

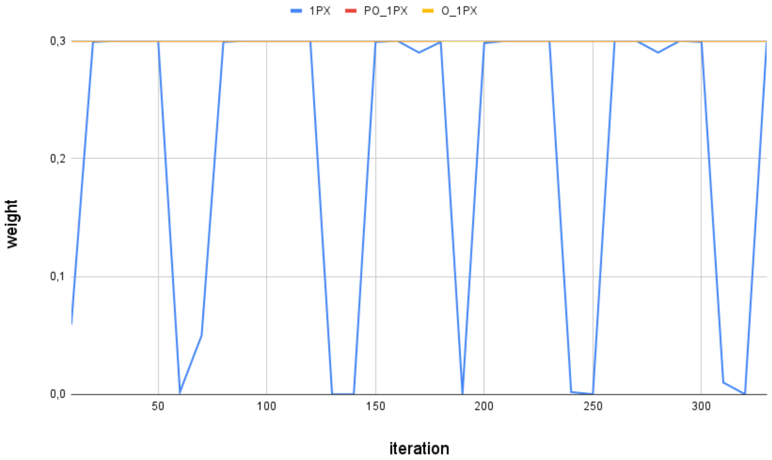
avg: 1.95%
min: 0.78%
max: 3.57%

GA with adaptation

Leading Crossover Operator: 1PX

avg: 2.06%
min: 0.83%
max: 3.88%

Dynamics of crossover weights during GA iterations



The classic restarting rule is used.

Results with IRACE

30 instances, $n = 50$

Parameter values of genetic algorithm found by IRACE

Parameter name	Parameter value
k	244
n_e	146
P_{IPRand}	0.43
$Selection$	Ranking
P_{Cross}	0.7
$Crossover$	1PX
P_{Mut}	0.63
$Mutation$	Exchange (swap)

Relative deviation of objective function found by the GA from the lower bound

avg: 1.99%
min: 0.82%
max: 3.86%

Results for testing set

Relative deviation of objective function found by the GA with IRACE parameters from the lower bound

avg: 1.8%
min: 0.76%
max: 3.22%

Relative deviation of objective function found by the GA with Adaptation from the lower bound

avg: 1.73%
min: 0.66%
max: 3.17%

Conclusions and Further Research

We recommend

- Apply modern packages for tuning numeric parameters.
- Apply adaptation for operators.
- Apply restarting rule for preventing premature convergence.

Further Research

- Use scramble mutation.
- Apply preprocessing packages for genetic algorithm with adaptation.

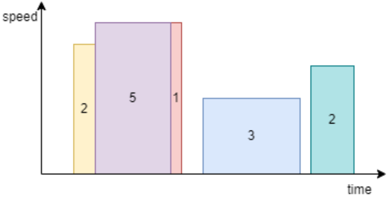
Thank you for your attention!

Problem 2

$m = 1$, the jobs have release dates and deadlines.

The objective is to find a feasible schedule that minimizes the total energy consumption.

Solution



Lower Bound