Конструктивные и эволюционные алгоритмы для составления расписаний с учетом расхода энергии при малом числе процессоров

М.Ю. Сахно

Омский филиал института математики
им. С.Л. Соболева СО РАН
Исследование выполнено за счет гранта Российского научного фонда № 22-71-10015, https://rscf.ru/project/22-71-10015.

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)
- Doerr & Ghannane, & Ibn Brahim: Runtime Analysis for Permutation-based Evolutionary Algorithms (2024)

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

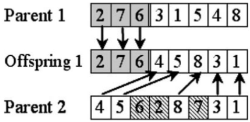


Figure: One Point Crossover (1PX)

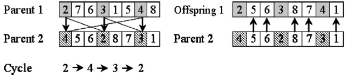


Figure: Cycle Crossover (CX)

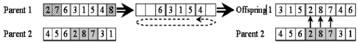


Figure: Order Crossover (OX)

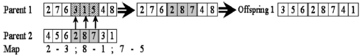


Figure: Partially Mapped Crossover (PMX)

Mutation Operators

Exchange (swap) mutation

3	8	4	1	5	2	7	6	⟹	3	8	7	1	5	2	4	6

Shift (insert) mutation

3	8	4	1	5	2	7	6	⟹	3	8	4	5	2	7	1	6

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).

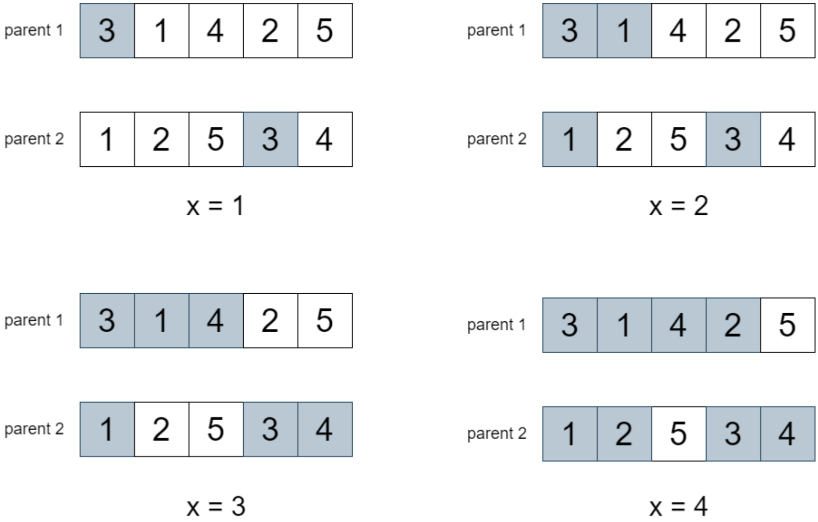
Optimal recombination may be considered as a best-improving move in a large neighbourhood defined by two parent solutions.

Property

Partial order given by the permutation.

Optimized Crossovers

One Point Crossover (1PX)



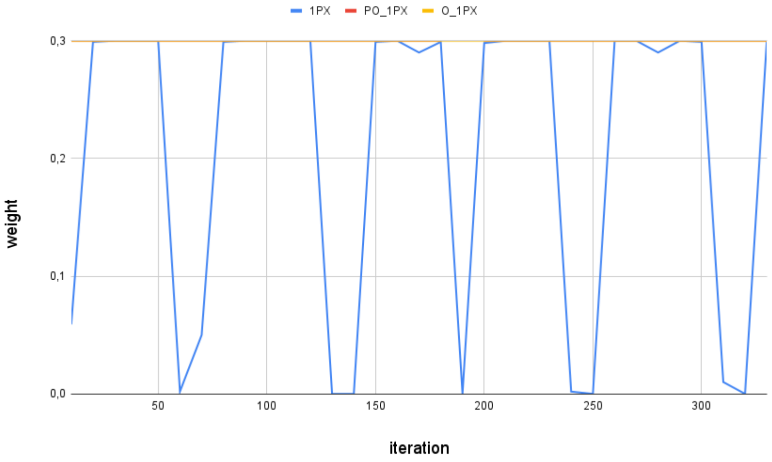
Adaptive Technique

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:

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

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

Dynamics of crossover weights during GA iterations



The classic restarting rule is used.

Scramble Mutation Scheme

1. Randomly choose n_p from Poisson distribution with λ_p.
2. Apply operator Mut for the given genotype n_p times.

Parameter auto-tuning

Parameter name	Parameter description
k	population size
n_e	number of elites
P_{IPRand}	probability of generating a genotype randomlly
$Selection$	selection operator
P_{Cross}	probability of applying the crossover operator
$Crossover$	crossover operator
P_{Mut}	probability of applying the mutation operator
$Mutation$	mutation operator
w_2, w_3, λ	parameters of adaptive technique
λ_p	lambda for Poisson distribution

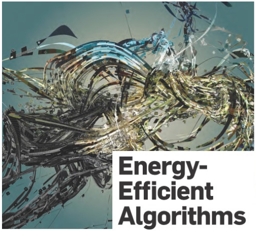
Speed Scaling Scheduling

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
r_j is the release date of the job j
d_j is the deadline for the job j

Parameters

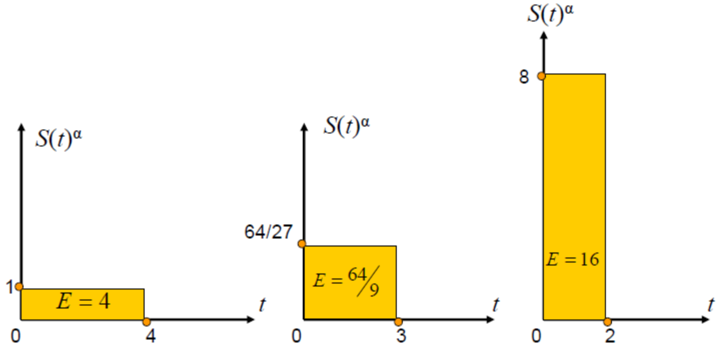
Non-preemptive instances arise in systems with distributed memory.



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 $\alpha \leq 3$).

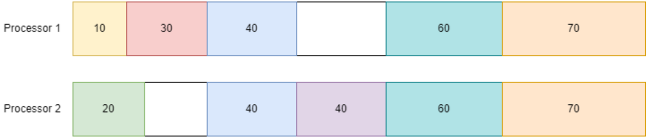
It is supposed that a continuous spectrum of processor speeds is available.



Problem 1

$m = 2$, E is the energy budget, $\sum C_j$ is the criteria.

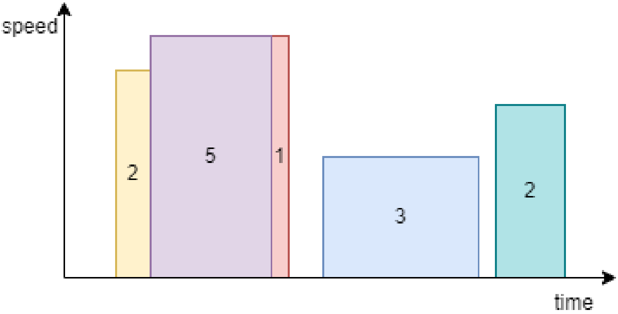
Solution Example



Problem 2

$m = 1$, E is the criteria.

Solution Example



Results of GA for Problem 1

	GA_{rand}	GA_{adapt_rand}	GA_{opt}	GA_{adapt_opt}	GR_{LI}
avg	2.03%	2.06%	2.11%	1.95%	4.56%
min	0.83%	0.83%	0.87%	0.78%	1.67%
max	3.83%	3.88%	3.69%	3.57%	7.74%

Table: Relative deviations of results from the lower bound for algorithms

	GA_{rand}	GA_{adapt_rand}	GA_{adapt_opt}
avg	1.99%	2.05%	1.94%
min	0.82%	0.83%	0.81%
max	3.86%	3.76%	3.63%

Table: Relative deviations of results from the lower bound for algorithms with parameters found by IRACE package

	$GA_{rand\_poisson}$	$GA_{adapt\_rand\_poisson}$	$GA_{adapt\_opt\_poisson}$
avg	1.96%	1.96%	1.95%
min	0.83%	0.86%	0.8%
max	3.72%	3.57%	3.63%

Table: Relative deviations of results from the lower bound for algorithms with scramble mutation operator

Results of GA for Problem 2

	GA_{rand_sc}	$GA_{adapt\_rand\_sc}$	GA_{adapt_opt}	$GA_{adapt\_opt\_sc}$
avg	0.00%	0.05%	0.01%	0.01%
min	0.00%	0.00%	0.00%	0.00%
max	0.04%	0.44%	0.27%	0.1%

Table: Relative deviations of results from the lower bound for algorithms

	GA_{rand_sc}	$GA_{adapt\_rand\_sc}$	GA_{adapt_opt}	$GA_{adapt\_opt\_sc}$
avg	0.00%	0.25%	0.00%	0.37%
min	0.0%	0.00%	0.0%	0.0%
max	0.0%	7.58%	0.0%	11.22%

Table: Relative deviations of results from the optimal solution for algorithms on the special test series

Scheduling with Predictions for Jobs Processing Times

Formulation

1 processor
x_j is predicted value of processing time of job j
$\sum C_j$ is criteria

Offline Algorithm

The Shortest Remaining Predicted Processing Time First (SRPPT) algorithm

Online Algorithm

The Round Robin algorithm (RRA)

Preferential Algorithm

SRPPT + Round Robin Algorithm (PA) [Bampis E., Dogeas K., Kononov A., Lucarelli G., Pascual F., 2022]

Algorithms Comparison Results

30 instances

	$SRPPT$	RRA	PA
avg	4.84%	73.97%	34.16%
min	1.05%	67.98%	28.25%
max	11.46%	79.65%	43.38%

Table: Relative deviations of results from the optimal solution for algorithms

Thank you for your attention!