Empirical Convergence Analysis Of Genetic Algorithm For Solving Unit Commitment Problem

Domen Butala

Coauthors:
doc. dr. Dejan Veluščček
doc. dr. Gregor Papa

Ljubljana, September 13th, 2014

Introduction

- Power system

Introduction

- Power system
- Unit Commitment problem?

Introduction

- Power system
- Unit Commitment problem?
- Motivation for an optimization approach

Introduction

- Power system
- Unit Commitment problem?
- Motivation for an optimization approach
- Techniques as MIP, MILP, LR, Benders Decomposition, Dynamic Programming, ...

An Upper Bound on the Convergence Speed

Theorem

[1] Let the size of population of the GA be $n \geq 1$, coding length $l > 1$, mutation probability $0 < p_m \leq \frac{1}{2}$ and let $\{\vec{X}_t, t \geq 0\}$ be the Markov chain population, $\pi^{(t)}$ distribution of t^{th} generation of $\vec{X}_t$ and π be the stationary distribution. Then it holds

$$||\pi^{(k)} - \pi|| \leq (1 - (2p_m)^{nl})^k.$$

Convergence of Homogenous Algorithm

Theorem

[2] Let $a, b, c > 0$ be constants and i intensity perturbations of algorithm. If it holds

$$m > \frac{an + c(n-1)\Delta^{\otimes}}{\min(a, b/2, c\delta)}, \qquad (1)$$

then

$$\forall x \in S^N : \qquad \lim_{i\to\infty} \lim_{t\to\infty} P([X_t^i] \subset f^* | X_0^i = x) = 1.$$

Combination of Both Approaches

Idea, to get the best algorithm possible, is to set a sequence of parameters $\{(n_t, p_m(t)), t \geq\}$ that it holds $n_t < n_{t+1}$ and $p_m(t) > p_m(t+1)$.

A Genetic algorithm set like this could be called a variable-structure GA [1].

Problem formulation

$$\min_{x_{i,t}^{type}} \left\{ \sum_{t=1}^{T} \sum_{i=1}^{n} (mp_{i,t} x_{i,t}^{type} + \max\{s_{i,t} - s_{i,t-1}, 0\} sc_i) \right\}$$

$$\sum_{i=1}^{n} x_{i,t}^{type} \geq PDP_t(price), \forall t \tag{2}$$

$$s_{i,t} = \left\{ \begin{array}{ll} 1, & \text{"if } x_{i,t}^{type} > 0, \\ 0, & \text{otherwise.} \end{array} \right\}, \forall t,\ i \tag{3}$$

$$st_{i,t} = (-1)^{1-s_{i,t}} \sum 1_{\left\{ \substack{I=[t-a,t+b] \wedge a,b \geq 0: \\ s_{i,t}=s_{i,\bar{t}} \forall \bar{t} \in I \wedge s_{i,t-a-1}=s_{i,t+b+1}=1-s_{i,t}} \right\}} \tag{4}$$

$$st_{i,t} \geq tup_i \vee st_{i,t} \leq -tdown_i, \forall t,\ i \tag{5}$$

$$x_{i,t} = xmax_{i,t} \tag{6}$$

Algorithm

1: $t = 0$
2: $P(t) = \text{SetInitialPopulation}(P)$

Algorithm

$t = 0$
$P(t) = \text{SetInitialPopulation}(P)$
$\text{Evaluate}(P(t))$
while not EndingCondition() **do**
    $t += 1$
    $P(t) = \text{Selection}(P(t-1))$

Algorithm

```
t = 0
P(t) = SetInitialPopulation(P)
Evaluate(P(t))
while not EndingCondition() do
    t+ = 1
    P(t) = Selection(P(t − 1))
    P(t) = Crossover(P(t))
```

Algorithm

$t = 0$
$P(t) = \text{SetInitialPopulation}(P)$
$\text{Evaluate}(P(t))$
while not EndingCondition() **do**
    $t+ = 1$
    $P(t) = \text{Selection}(P(t-1))$
    $P(t) = \text{Crossover}(P(t))$
    $P(t) = \text{Mutation}(P(t))$

Algorithm

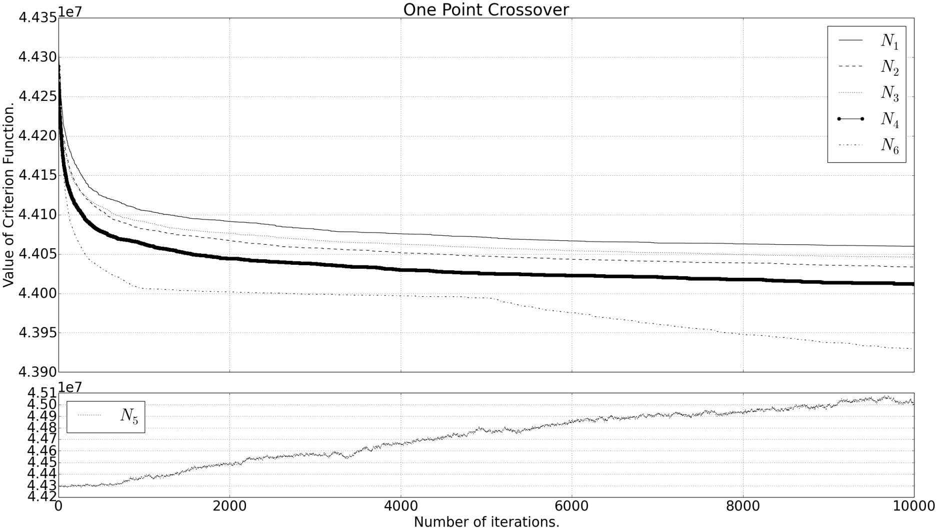
$t = 0$
$P(t) = \text{SetInitialPopulation}(P)$
$\text{Evaluate}(P(t))$
while not EndingCondition() **do**
    $t+ = 1$
    $P(t) = \text{Selection}(P(t-1))$
    $P(t) = \text{Crossover}(P(t))$
    $P(t) = \text{Mutation}(P(t))$
    $\text{Evaluate}(P(t))$
end while

Results

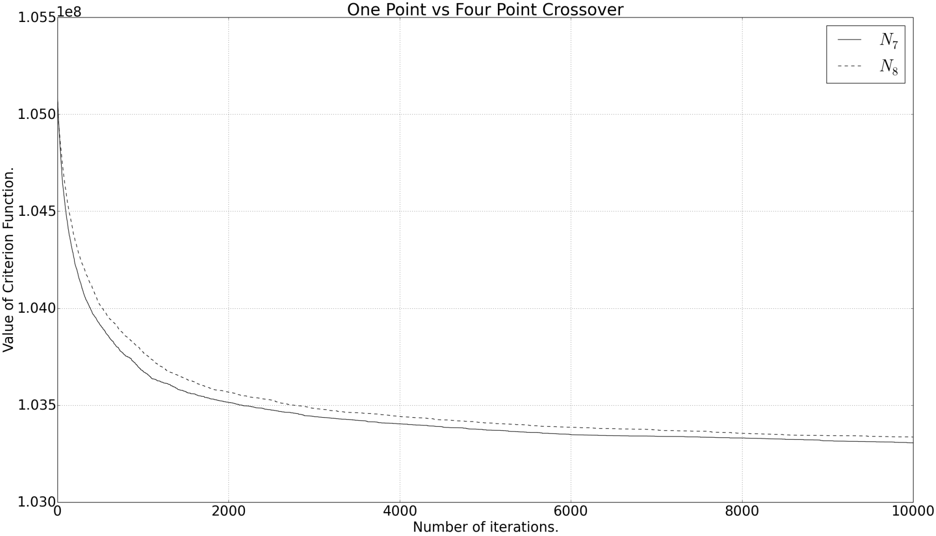
Table 1. Parameter settings.

Parameters / Settings	N_1	N_2	N_3	N_4	N_5	N_6
Iterations	10,000	10,000	10,000	10,000	10,000	10,000
Population size	30	30	30	60	30	60
Elitism	4	4	4	4	0	4
Crossover	OPC	OPC	OPC	OPC	OPC	OPC
Crossover Probability	50%	75%	50%	50%	50%	vary
Mutation Probability						
- population	25%	25%	40%	25%	25%	vary
- individual	20%	20%	25%	20%	20%	vary

One-point crossover



Multi-point crossover



Appendix

Algorithm was implemented in the programming language Python.

- Numpy
- Cython
- Numba

On some parts of the code performance comparisons were made to implementations in other languages (R, Matlab and C#).

- **Domen Butala**
  Financial Mathematics, Faculty of Mathematics and Physics, Ljubljana, Slovenia
  domen.butala@yahoo.com
- **Dejan Veluščček**
  Department of Mathematics, Faculty of Mathematics and Physics, Ljubljana, Slovenia
  dejan.veluscek@fmf.uni-lj.si
- **Gregor Papa**
  Computer Systems Department, Jožef Stefan Institute, Ljubljana, Slovenia
  gregor.papa@ijs.si

Y. Gao, *An Upper Bound on the Convergence Rates of Canonical Genetic Algorithms*. Complexity International, Vol. 5, 1998.

R. Cerf, *Asymptotic Convergence of Genetic Algorithms.* CNRS, Université d'Orsay, Paris, 1997.