AN ADAPTIVE QUANTUM-BASED MULTI-OBJECTIVE EVOLUTIONARY ALGORITHM FOR EFFICIENT TASK ASSIGNMENT IN DISTRIBUTED SYSTEMS

JERZY BALICKI

Overview

- Adaptive quantum-based multi-objective evolutionary(AQMEA) algorithm is proposed in this paper.
- By utilizing the benefits of a quantum-inspired evolutionary algorithm, it improves closeness to the Pareto-optimal front while preserving variety (QEA).

Related Work

- A unique quantum-inspired multi-objective evolutionary algorithm (QMEA) is described in earlier papers.
- Proved to be better than conventional genetic algorithms for multiple objective optimization problems.
- QMEA finds solutions close to the Pareto-optimal front, of higher quality than NSGA-II.
- NSGA-II is a notable, fast sorting and elite multi objective conventional genetic algorithm

Problem Statement

Multi-criterion problem of task assignment

In order to reduce

- Workload of a bottleneck computer
- Cost of the system

As a result, the distributed system's dependability improves.

Terminology

- What is Multi-Criterion Programming?
- What is Q-bit?
- What is Evolutionary Genetic Algorithm
- What is Pareto Optimal front?

What is Multi-Criterion Programming?

Multi-Criterion Programming is an Optimization Problem that involves multiple objective functions .

In Distributed Systems if we consider the case of task scheduling the following objectives arises

- Z max - A workload of a bottleneck computer

$$Z_{\max}(x) = \max_{i \in \overline{1,I}} \left\{ \sum_{j=1}^{J} \sum_{v=1}^{V} t_{vj} x_{vi}^{m} x_{ij}^{\pi} + \sum_{v=1}^{V} \sum_{\substack{u=1 \\ u \neq v}}^{V} \sum_{i=1}^{I} \sum_{\substack{k=1 \\ k \neq i}}^{I} \tau_{vuik} x_{vi}^{m} x_{uk}^{m} \right. \quad (10)$$

- C – The cost of the system is minimized

$$C(x) = \sum_{i=1}^{I} \sum_{j=1}^{J} \kappa_j x_{ij}^{\pi} \quad (11)$$

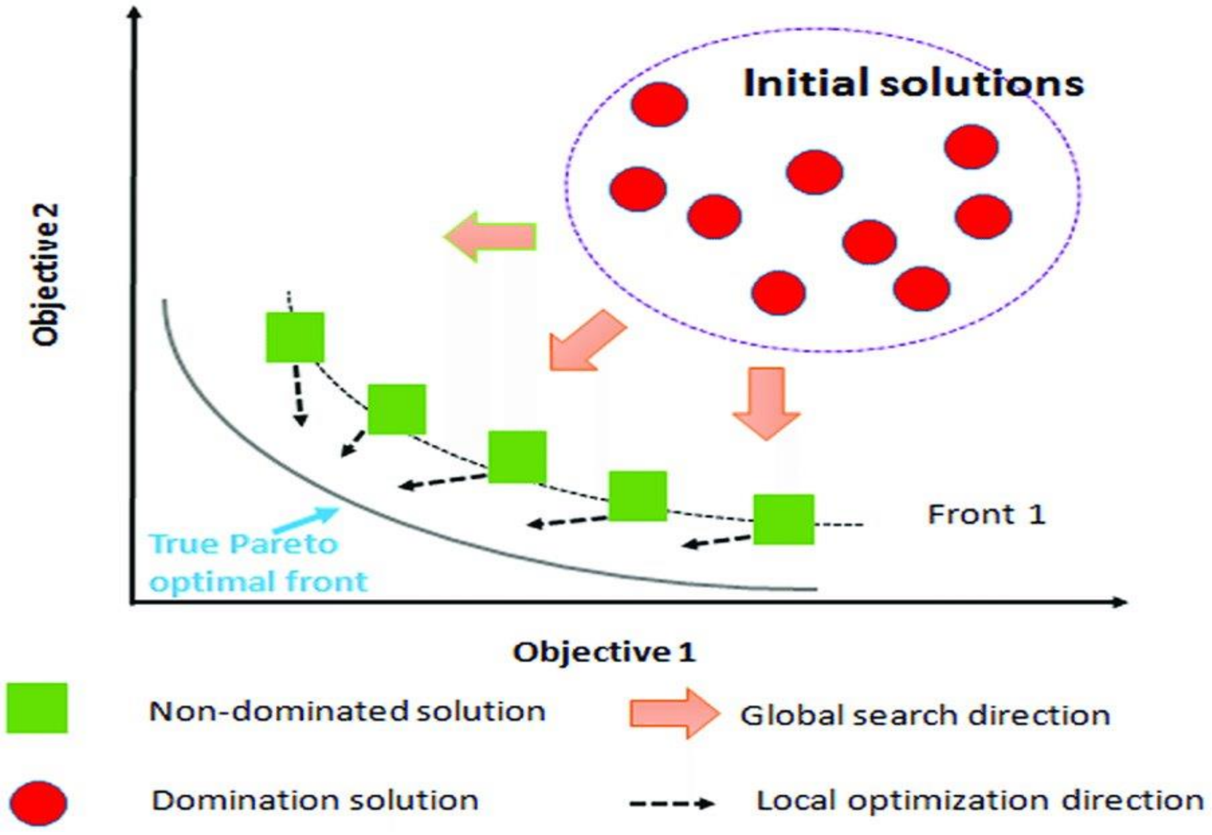
where κ_j corresponds to the cost of the computer π_j.

- What is Q-bit ?

AQMEA utilizes a new representation, called a Q-bit, for the probabilistic representation that is based on the concept of qubits. A qubit is a two-layer quantum system that can be modeled as the Hilbert space H2 with the give base . $B \{|0\rangle, |1\rangle\}$

$$Q_m = \alpha_m|0\rangle + \beta_m|1\rangle \text{ , where}$$

M is the number of the gene in the chromosome



Pareto efficiency is believed to occur when it is impossible to benefit one party without harming another.

What is Evolutionary Genetic algorithm ?

Genetic algorithms are often employed to develop high-quality solutions to optimization and search problems by utilizing biologically inspired operators including mutation, crossover, and selection.

Evolutionary Genetic Algorithm

Population

randomly generate individuals, with the population in each iteration called a *generation*.

Fitness Calculation

Its is the value of the objective functions in the optimization problem

Convergence check

Selection

Crossover

Mutation

Algorithm proposed - AQMEA

$$p_s(Q^i) = \frac{\sum_{j=1}^{\lambda} f(x^{ij})}{\sum_{x^{ij} \in P(t)} f(x^{ij})}, \quad i = \overline{1, \zeta} \qquad (6)$$

Roulette Wheel Selection, where F is fitness function

1. **BEGIN**
2. t:=0, t – the number of population
3. set ζ the size of Q-population Θ, L size of binary population $\boldsymbol{P}$, $L = \lambda\,\zeta$, for the given sampling parameter λ
4. p_m:=1/($M\,\zeta$), M – the length of x
5. generate an initial population $\Theta(0)$ *and* $\boldsymbol{P}(0)$,
6. calculate non-dominated ranks $r(x)$ and fitness $f(x), x \in \boldsymbol{P}(t)$
7. *finish*:=FALSE
8. WHILE NOT *finish* DO
9. BEGIN /* new populations Θ and $\boldsymbol{P}$ */
10. t:= t+1, $\boldsymbol{P}(t) := \varnothing$, $\Theta(t) := \varnothing$
11. calculate the selection probabilities $p_s(x)$, $x \in \boldsymbol{P}(t-1)$ by (6)
12. FOR ζ /2 DO
13. BEGIN /* reproduction cycle */
14. 2WT-selection of a potential parent pair $\{\boldsymbol{a},\boldsymbol{b}\}$ from $\Theta(t\text{-}1)$
15. Q-crossover of a parent pair $\{\boldsymbol{a},\boldsymbol{b}\}$ with the adaptive crossover rate p_c, $p_c := e^{-t/T_{max}} - 0.1$
16. Q-mutation of an offspring pair $\{\boldsymbol{a}',\boldsymbol{b}'\}$ with the adaptive mutation rate, $p_m := \dfrac{t}{\zeta M T_{max}}$
17. $\Theta(t) := \Theta(t) \cup \{\boldsymbol{a}',\boldsymbol{b}'\}$
18. END
19. generate $\boldsymbol{P}(t)$ by observing $\boldsymbol{\Theta}(t)$ λ times
20. calculate ranks $r(x)$ and fitness $f(x), x \in \boldsymbol{P}(t)$
21. IF ($\boldsymbol{P}(t)$ converges OR $t \geq T_{max}$) THEN *finish*:=TRUE
22. END
23. END

Population

- Chromosome is a set of parameters which define a proposed solution to the problem that the genetic algorithm is trying to solve.

- Initial population is generated from random numbers αm € [-1;1] and calculation βm from the normalization formula.

$$Q_m = \alpha_m|0\rangle \oplus \beta_m|0\rangle,$$

$$|\alpha_m|^2 + |\beta_m|^2 = 1,$$

Normalization

$$Q = \begin{bmatrix} \alpha_1 & \dots & \alpha_m & \dots & \alpha_M \\ \beta_1 & \dots & \beta_m & \dots & \beta_M \end{bmatrix}$$

Q – Chromosomes Matrix

Consider the chromosome matrix below

$$Q = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{7}} & \frac{1}{\sqrt{2}} & \frac{1}{2} \\ \frac{-\sqrt{2}}{\sqrt{3}} & \frac{2}{\sqrt{5}} & \frac{\sqrt{3}}{\sqrt{7}} & \frac{1}{\sqrt{2}} & \frac{\sqrt{3}}{2} \end{bmatrix}$$

$$Q = \begin{bmatrix} \alpha_1 & \dots & \alpha_m & \dots & \alpha_M \\ \beta_1 & \dots & \beta_m & \dots & \beta_M \end{bmatrix}$$

Adaptive quantum-based multi-objective evolutionary(AQMEA) algorithm is proposed in this paper.

By utilizing the benefits of a quantum-inspired evolutionary algorithm, it improves closeness to the Pareto-optimal front while preserving variety (QEA).

N_1 N_2 N_1 N_2 N_1 N_2 N_1 N_2 N_1 N_2

(0 , 1 , 0 , 1 , 1 , 0 , 1 , 0 , 0 , 1)

T1 T2 T3 π1 π 2

Q-chromosomes matrix

The first criterion is the workload of the bottleneck computer for the allocation x, and its values are provided by the subsequent formula

Zmax – a workload of a bottleneck computer

$$Z_{\max}(x) = \max_{i \in \overline{1,I}} \left\{ \sum_{j=1}^{J} \sum_{v=1}^{V} t_{vj} x_{vi}^{m} x_{ij}^{\pi} + \sum_{v=1}^{V} \sum_{\substack{u=1 \\ u \neq v}}^{V} \sum_{\substack{i=1 \\ k \neq i}}^{I} \sum_{k=1}^{I} \tau_{vuik} x_{vi}^{m} x_{uk}^{m} \right. \quad (10)$$

$$(x^m, x^\pi) = [x_{11}^m, \ldots, x_{1I}^m, \ldots, x_{vi}^m, \ldots, x_{VI}^m, x_{11}^\pi, \ldots, x_{1J}^\pi, \ldots, x_{ij}^\pi, \ldots, x_{I1}^\pi, \ldots, x_{Ij}^\pi, \ldots, x_{IJ}^\pi]^T.$$

$$x_{ij}^{\pi} = \begin{cases} 1 \text{ if } \pi_j \text{ is assigned to the } w_i, \\ 0 \text{ in the other case.} \end{cases}$$

$$x_{vi}^{m} = \begin{cases} 1 \text{ if task } T_v \text{ is assigned to } w_i, \\ 0 \text{ in the other case,} \end{cases}$$

$\Pi = \{\pi_1, \ldots, \pi_j, \ldots, \pi_J\}$ - the set of available computer sorts,

$\{T_1, \ldots, T_v, \ldots, T_V\}$ - the set of parallel performing tasks,

$W = \{w_1, \ldots, w_i, \ldots, w_I\}$ - the se of the processing nodes,

$N^m = [N_1, \ldots, N_v, \ldots, N_V]^T$,

N_v – number of the vth module in the line for its dedicated computer,

t_{vj} - the overhead performing time of the task T_v by the computer π_j.

τ_{vu} – the total communication time between the task T_v and the T_u,

$z_1, \ldots, z_r, \ldots, z_R$ - memories available in the system,

d_{jr} - the capacity of memory z_r in the workstation π_j,

κ_j - the cost of the computer π_j,

ϑ_j - the numerical performance of the computer π_j for the given benchmark, The second measure of the task assignment is a cost of computers that is calculated, as below:

C – the cost of system are minimized

$$C(x) = \sum_{i=1}^{I} \sum_{j=1}^{J} \kappa_j x_{ij}^{\pi} \qquad (11)$$

where κ_j corresponds to the cost of the computer π_j.

Selection, mutation and crossover

SELECTION

- Roulette Wheel Selection (the probability of choosing an individual for breeding of the next generation, the better the fitness is, the higher chance for that individual to be chosen,
- where ***f*** is fitness function

$$p_s(Q^i) = \frac{\sum_{j=1}^{\lambda} f(x^{ij})}{\sum_{x^{ij} \in P(t)} f(x^{ij})}, \quad i = \overline{1,\zeta} \qquad (6)$$

crossover probability and the mutation rate are changed in an adaptive way. i.e Tmax is maximum population in the generatuion

CROSSOVER

- Crossover is carried out on quantum chromosomes instead of binary chromosomes as at classical genetic algorithm

Chromosome1	11011|00100110110
Chromosome2	11011|11000011110
Offspring1	11011|11000011110
Offspring2	11011|00100110110

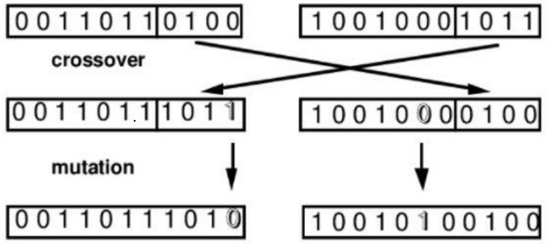
Single Point Crossover

$$p_c = e^{-t/T_{max}}$$

$$p_m := \frac{t}{\zeta M T_{max}}$$

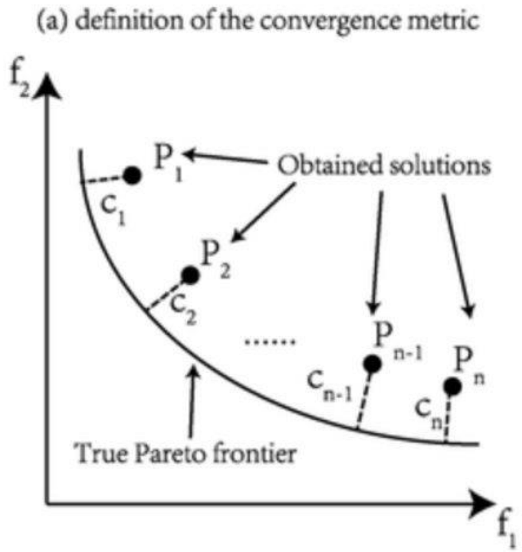
MUTATION

- Mutation may be defined as a small random tweak in the chromosome, to get a new solution. It is used to maintain and introduce diversity in the genetic population and is usually applied with a low probability – pm.



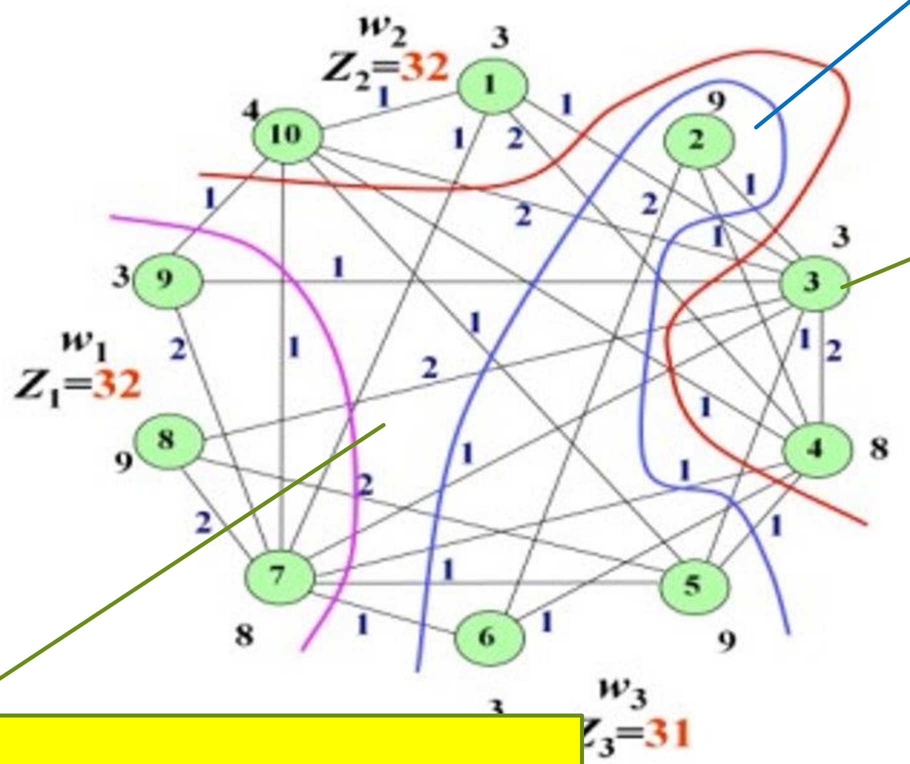
Justification – Results of previous algorithm

- Previously proposed algorithms have large pareto set convergence when the number or tasks, number of nodes and number of computer types are increased.
- Adaptative multi-chromosome evolutionary algorithm (AMEA) and Adaptative multi-chromosome evolutionary algorithm (AMEA)+
- For example, for 12 decision variables, the Pareto set obtaining is 1.8% for the AMEA+ , 3.4% for the AMEA .
- And for 80 decision variables, the Pareto set obtaining is 16.7% for the AMEA+ , 18.4% for the AMEA



Justification – Results from present paper

Load Balancing performed for 10 tasks and 3 Nodes



2nd Task computation time

Green circle represents the 10 tasks , marked my task number

total communication time between the task T9 and T3

$$Z_{\max}(x) = \max_{i \in \overline{1,I}} \left\{ \sum_{j=1}^{J} \sum_{v=1}^{V} t_{vj} x_{vi}^{m} x_{ij}^{\pi} + \sum_{v=1}^{V} \sum_{\substack{u=1 \\ u \neq v}}^{V} \sum_{\substack{i=1 \\ k \neq i}}^{I} \sum_{k=1}^{I} \tau_{vuik} x_{vi}^{m} x_{uk}^{m} \right. \quad (10)$$

- W1, hhh W2 , W3 are processing nodes
- Z1, Z2, Z3 are workloads of the nodes

Limitations

- AQMEA doesn’t take into account of repair and recovery times for failed computer during task processing.
- Proposed algorithm allocates tasks to computers only based on the fact which failures are least likely to occur during the execution

Future work

- The development of the negative selection algorithm (NSA) can be introduced to increase the quality of the result.
- In such method, the present population prefers infeasible solutions that are comparable to feasible ones. The emphasis is on improving the fitness of relevant infeasible solutions.

Conclusion

- The quantum-based adaptive evolutionary algorithm AQMEA is suggested to identify optimal solutions.
- It is a sophisticated approach for determining the Pareto-optimal job allocation problem while maximizing system dependability and distributed system performance.
- The workload of the bottleneck computer and the cost of computers are minimized and in contrast, the reliability of the distributed system is maximized.

References

- Warsaw (1986) 2. Balicki, J.: Tabu Programming for Multi objective Optimization Problems, International Journal of Computer Science and Network Security, 7, 44--50 (2007)
- Han, Kuk-Hyun, and Jong-Hwan Kim. "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization." IEEE transactions on evolutionary computation 6.6 (2002): 580-593.
- Balicki, Jerzy. "Immune systems in multi-criterion evolutionary algorithm for task assignments in distributed computer system." International Atlantic Web Intelligence Conference. Springer, Berlin, Heidelberg, 2005.
- Balicki, J.: Tabu Programming for Multi objective Optimization Problems, International Journal of Computer Science and Network Security, 7, 44--50 (2007)
- Cho, Jin-Hee, et al. "A survey on modeling and optimizing multi-objective systems." IEEE Communications Surveys & Tutorials 19.3 (2017): 1867-1901.
- Han, Kuk-Hyun, and Jong-Hwan Kim. "On setting the parameters of quantum-inspired evolutionary algorithm for practical application." The 2003 Congress on Evolutionary Computation, 2003. CEC'03.. Vol. 1. IEEE, 2003.

Thank You

Vaishnavi Sarikonda