On Swarms of Particles

Alexander G. Ororbia II

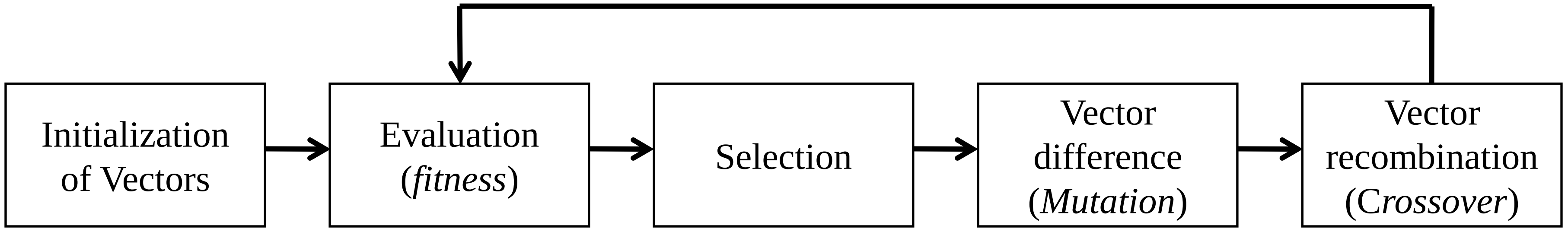
Biologically-Inspired Intelligent Systems

CSCI-633

2/22/2024

Metaheuristic: Differential Evolution (DE)

- Vector-based (population-based) algorithm; Storn & Price (1996/1997)
    - Viewed as self-organizing system
    - Individuals ‘evolve’ by recombination w/ other individuals & differentials between other individuals
- Devised for continuous search spaces, derivative-free
- No encoding/decoding required – real numbers are now solutions/chromosomes
- *DE/rand/1/bin*



Meta-parameter Selection

- Extensive work shows that meta-params/hyper-params should be tuned to problem
- DE is most sensitive to scale factor α, with $\alpha \in [0.4, 0.95]$ an empirically good range with a starting point of $\alpha \in [0.7, 0.9]$
- $C_r \in [0.1, 0.8]$ an empirically good range w/ $C_r = 0.5$ as a starting point
- For population size *N*, value should reflect dimensionality *d* of the problem, so something such as *N* = 5*D* (or 10*D*) → issues for high-dimensional problems
  - Can start with fixed value as starting point: *N* = 40 or 100

Some Convergence Thoughts / Issues

- Xue et al [10]* showed that λ should be large to yield better convergence
- Zaharie [13,14] – to avoid premature convergence (for any population-based algorithm), must maintain good degree of diversity [1]
  - Generally analyze/characterize the variance of DE variants (usually w/o selection) → generally lead to conclusions about meta-params such as α and how they affect variance of population solutions
- In general: when *var*(P) is decreasing/going down, DE is converging (or when *var*(P) → 0, DE has converged)
  - This convergence may be premature (i.e., not a global optima)
- ***<u>Issue</u>***: Population (P) diversity also depends on initial P
- ***<u>Issue</u>***: combinatorial problems – difficult to say if DE works well given how hard it is to discretize differential ops, etc.

* Reference/citation numbers match those in textbook ;-)

Implemented / Used In:

- **Knapsack Problem**

  - KRAUSE, Jonas; Parpinelli, R. S.; Lopes, H.S. **Proposta de um algoritmo inspirado em Evolução Diferencial aplicado ao Problema Multidimensional da Mochila**, 2012, Curitiba. Anais do Encontro Nacional de Inteligência Artificial – ENIA.

  - KRAUSE, Jonas; Cordeiro, J.A.; Lopes, H.S.. **Comparação de Métodos de Computação Evolucionária para o Problema da Mochila Multidimensional.** In: H.S. Lopes; L.C.A. Rodrigues; M.T.A. Steiner. (Org.). Meta-Heurísticas em Pesquisa Operacional. 1ed.Curitiba: Omnipax, 2013, p. 87-98.

  - KRAUSE, Jonas; Lopes, H.S. **A comparison of differential evolution algorithm with binary and continuous encoding for the MKP.** In: BRICS - Conference on Computational Intelligence, 2014, Recife. Proceedings of BRICS-CCI, 2013.

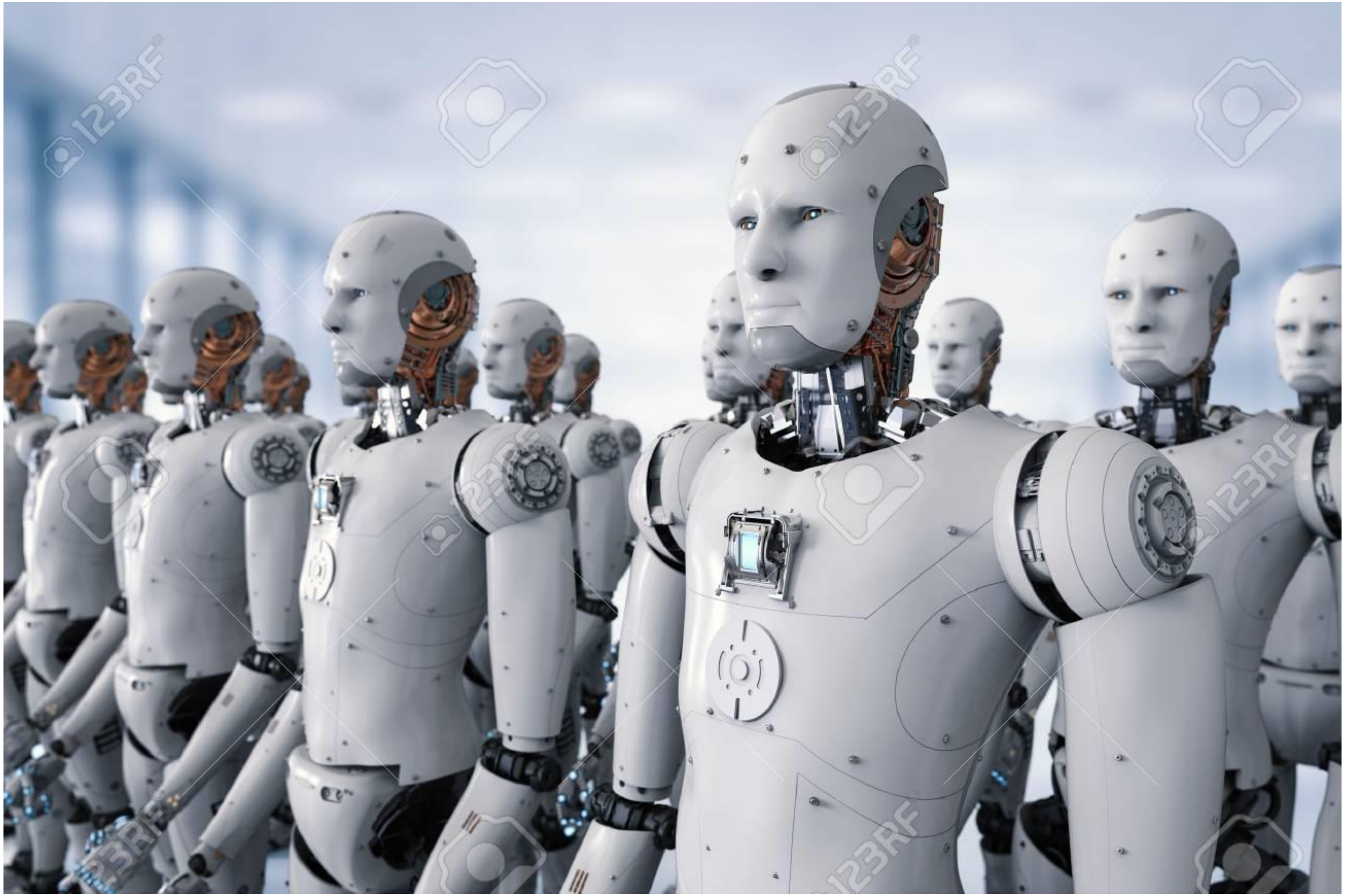
- **Scheduling Problem**

  - KRAUSE, Jonas; Sieczka, E.; Lopes, H.S. **Differential Evolution Variants and MILP for the Pipeline Network Schedule Optimization Problem.** In: LA-CCI - Congress on Computational Intelligence, 2015, Curitiba.

Swarm Intelligence

- Uses real number randomness & global communication (instead of mutation/crossover)
- Easier to implement – (also) no encoding/decoding needed

- **<u>Operation</u>** – adjust piecewise paths of individual agents (quasi-stochastic manipulation of positional vectors)



Roger, roger!

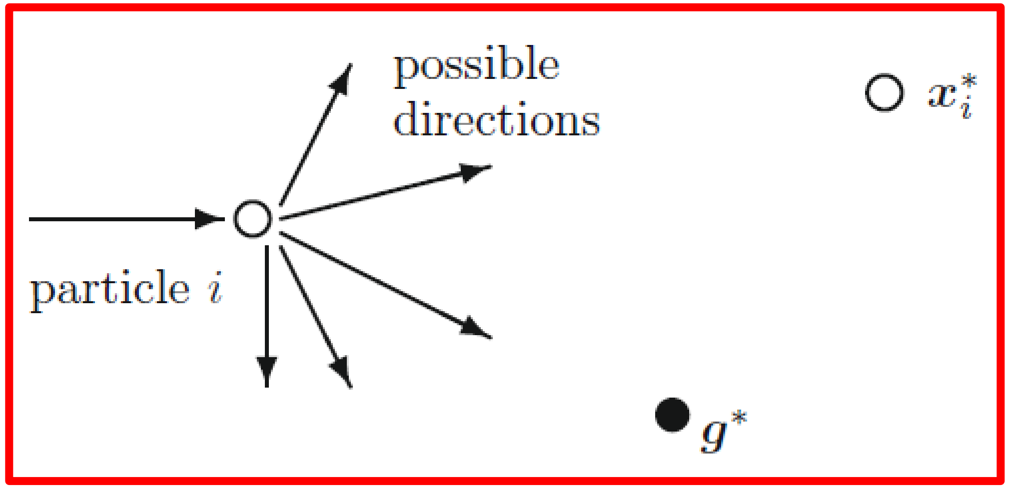
Particle Swarm Optimization (PSO)

- Agent *i* = “particle”, guided by stochastic & deterministic component (Kennedy & Eberhart, 1995)
  - Inspired by swarm/schooling behavior of fish and birds
- Attracted to:
  - Current global best location, $\boldsymbol{g}^*$ (social intelligence)
  - Current local best location, $\boldsymbol{x}_i^{*(t)}$ (cognitive intelligence)
- Has tendency to move randomly (injected noise)

- Using local (individual) best might be increasing diversity in solution quality

PSO Mechanics

- When particle finds better position (than in history), updates location for agent *i*
- At any *t*, a global best for *n* agents is tracked
  - Aim: find global best among current best solutions until objective no longer improves (or after iteration cutoff)



PSO Mechanics

Particle Swarm Optimization

Objective function $f(\boldsymbol{x})$, $\boldsymbol{x} = (x_1, ..., x_d)^T$
Initialize locations $\boldsymbol{x}_i$ and velocity $\boldsymbol{v}_i$ of n particles.
Find $\boldsymbol{g}^*$ from $\min\{f(\boldsymbol{x}_1), ..., f(\boldsymbol{x}_n)\}$ (at $t = 0$)
while (criterion)
    for loop over all n particles and all d dimensions
    Generate new velocity $\boldsymbol{v}_i^{t+1}$ using equation (7.1)
    Calculate new locations $\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \boldsymbol{v}_i^{t+1}$
    Evaluate objective functions at new locations $\boldsymbol{x}_i^{t+1}$
    Find the current best for each particle $\boldsymbol{x}_i^*$
    end for
    Find the current global best $\boldsymbol{g}^*$
    Update $t = t + 1$ (pseudo time or iteration counter)
end while
Output the final results $\boldsymbol{x}_i^*$ and $\boldsymbol{g}^*$

Questions?