**KEYWORDS** A

* Hill Climbing Approach - a form of local search, which means it focuses on finding the optimal solution by making incremental changes to an existing solution and then evaluating whether the new solution is better than the current one. [[1](https://cs.gmu.edu/~sean/book/metaheuristics/Essentials.pdf#page=11.09)]
* Metaheuristics - A metaheuristic is a higher level procedure or heuristic designed to find, generate, or select a lower level procedure or heuristic (partial search algorithm) that may provide a sufficiently good solution for an optimization problem.
* Intensification/ Exploitation - Intensification is a strategy that focuses on exploring the promising areas of the search space more thoroughly. It is often associated with exploitation, where the algorithm tries to refine the best solutions found so far. This strategy is useful for finding local optima and improving the quality of solutions. Intensification is typically achieved by guiding the search towards areas of the search space that have already yielded good solutions, with the hope of finding even better solutions in their vicinity.
* Diversification / Exploration – diversification is a strategy that aims to explore many different regions of the search space. It is often associated with exploration, where the algorithm tries to discover new, unexplored areas of the search space. This strategy is useful for escaping local optima and ensuring a good coverage of the search space. Diversification is typically achieved by introducing randomness into the search process or by penalizing the revisiting of previously explored areas.
* Local Optimum - A local optimum is an extrema (maximum or minimum) point of the objective function for a certain region of the input space. More formally, for the minimization case xlocal is a local minimum of the objective function f if:
  + f(x) ≥ f(xlocal) for all values of x in range [xlocal - ε, xlocal + ε].
* Neighborhood Modelling / Methods - strategy of defining a "neighborhood" for a given solution. This neighborhood consists of all possible solutions that can be reached by applying a small change or "move" to the current solution.
* Trajectory Population-Based Metaheuristics
  + Trajectory-Based Metaheuristics: These methods involve a single solution or trajectory through the solution space, iteratively improving or modifying the solution until an optimal or satisfactory solution is found. Examples include Simulated Annealing, Tabu Search, and GRASP.
  + Population-Based Metaheuristics: These algorithms use a population (or multiple solutions) simultaneously, allowing the algorithm to explore multiple regions of the solution space in parallel. The solutions evolve based on operations such as selection, recombination, and mutation. Genetic Algorithms and Ant-Colony Optimization are classic examples.
* Genetic Algorithm - population-based metaheuristic inspired by the principles of natural evolution and genetics. It iteratively evolves a population of solutions by selecting the fittest individuals, performing crossover (recombination), and applying random mutations. Over successive generations, the population tends to converge towards optimal or near-optimal solutions.
* Tabu Algorithm - trajectory-based metaheuristic that uses memory structures to escape local optima by "tabu-listing" recently visited solutions or moves, thus preventing cycles.
* Simulated Annealing - trajectory-based algorithm that probabilistically accepts worse solutions in the hopes of escaping local optima. This probability decreases as the "temperature" decreases over time.
* GRASP (Greedy Randomized Adaptive Search Procedure) - a two-phase, multi-start metaheuristic where each iteration consists of a construction phase (building a solution using a greedy randomized approach) and a local search phase (improving the solution by exploring its neighborhood).
* Construction-Based Methods - algorithms that iteratively build feasible solutions from scratch, typically using heuristic rules to guide the construction process. These methods focus on building a good initial solution quickly, which can then be improved through local search or other optimization techniques.
* Ant-Colony Algorithm - population-based metaheuristic inspired by the behavior of ants searching for food. In ACO, artificial "ants" construct solutions by moving through a solution space and depositing pheromones, which guide subsequent ants. This pheromone trail evolves over time, reinforcing paths that lead to good solutions while allowing suboptimal ones to fade, thus balancing exploration and exploitation.
* Lower Bound - the minimum value that an objective function can theoretically achieve.
* Perturbative-Local Search Method – involves applying small changes or "perturbations" to an existing solution and then performing local search in its neighborhood to find improved solutions. [[2](http://www.scholarpedia.org/article/Metaheuristics)]
* No-Free Lunch Theorem - no single optimization algorithm is universally effective for all problems.
* Roughness - the complexity and irregularity of the solution landscape. A "rough" landscape is marked by numerous local optima and abrupt changes, making it challenging for algorithms to find the global optimum.
* Bound

**CONTEXT** A

A routing problem is being addressed using metaheuristics. Agathe raises concerns about relying on simple heuristics and suggests exploring advanced methods like genetic algorithms and simulated annealing.

**PROBLEM STATEMENT** A

How can we select an effective metaheuristic that balances intensification and diversification, while also evaluating solution quality, generating representative problem instances, and fine-tuning algorithm parameters for optimal performance?

**CONSTRAINTS** A

* NP-Complete problem
* Computation time

**SOLUTION APPROACH / HYPOTHESIS** A

* Balance intensification and diversification
* Use genetic algorithm
* Test and compare metaheuristics approaches

**ACTION PLAN** A

* Study and compare metaheuristics algorithms – Deliverable for the prosit
* Propose two metaheuristics for the project – Deliverable for the project

**IMPORTANT LINKS** A

<https://klu.ai/glossary/metaheuristic>

**GENETIC ALGORITHM** A

Genetic Algorithms(GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random searches provided with historical data to direct the search into the region of better performance in solution space. They are commonly used to generate high-quality solutions for optimization problems and search problems.

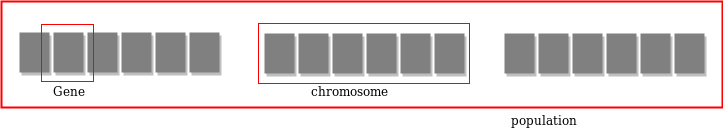
**Foundation of Genetic Algorithms**

Genetic algorithms are based on an analogy with the genetic structure and behavior of chromosomes of the population. Following is the foundation of GAs based on this analogy –

* Individuals in the population compete for resources and mate
* Those individuals who are successful (fittest) then mate to create more offspring than others
* Genes from the “fittest” parent propagate throughout the generation, that is sometimes parents create offspring which is better than either parent.
* Thus each successive generation is more suited for their environment.

**Search space**

The population of individuals are maintained within search space. Each individual represents a solution in search space for given problem. Each individual is coded as a finite length vector (analogous to chromosome) of components. These variable components are analogous to Genes. Thus a chromosome (individual) is composed of several genes (variable components).



**Fitness Score**

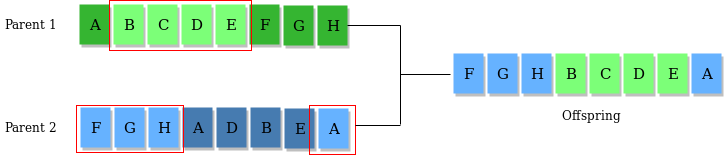
A Fitness Score is given to each individual which shows the ability of an individual to “compete”. The individual having optimal fitness score (or near optimal) are sought.

**Operators of Genetic Algorithms**

Once the initial generation is created, the algorithm evolves the generation using following operators –

1) Selection Operator: The idea is to give preference to the individuals with good fitness scores and allow them to pass their genes to successive generations.

2) Crossover Operator: This represents mating between individuals. Two individuals are selected using selection operator and crossover sites are chosen randomly. Then the genes at these crossover sites are exchanged thus creating a completely new individual (offspring). For example –



3) Mutation Operator: The key idea is to insert random genes in offspring to maintain the diversity in the population to avoid premature convergence. For example –



**Algorithm**

1) Randomly initialize populations p

2) Determine fitness of population

3) Until convergence repeat:

a) Select parents from population

b) Crossover and generate new population

c) Perform mutation on new population

d) Calculate fitness for new population

**ANT COLONY OPTIMIZATION** A

Ant Colony Optimization technique is purely inspired from the foraging behaviour of ant colonies.

# Not yet done