Software product line engineering is an approach for systematic reuse common set of features across a very large number of similar products. SPL engineering addresses well-known needs of software engineering such as, reducing the cost of development and maintenance, increasing the quality, and decreasing the time to market. Evaluating the reliability of an SPL is important, because some of its features are constantly reused. Hence, testing becomes necessary to avoid fault propagation to the derived products.

Testing an SPL is a difficult task due to the explosion of possible products that need to be tested as a result of the high number of possible combination features. The main concern is that it is not possible to test all the possible products because the resources for testing are usually limited. To tackle this problem, several approaches have been proposed to reduce the number of products to test e.g., combinatorial interaction testing (Pairwise interaction testing and T-wise testing).

The challenge is that computing all the t-wise interaction from a feature model with the presence of all the constraints is still a problem, especially for a large feature model such as the Linux kernel (over 11,000 features). Hence, we have to select a subset of the possible products. The random selection is very poor technique at finding solutions (in our case products) when those solutions occupy a very small part of the overall search space (in our case, feature model). The products may be found faster and more reliably if the search given some guidance. Meta-heuristic searches can provide this guidance in the form of the fitness function. The fitness information can be utilized by optimization algorithms, such as

- Hill Climbing (local search approach)
- Simulated Annealing (local search approach)
- Genetic algorithm (global search approach)
- Particles Swarm Optimization (PSO) (global search approach)