2

Meta-heuristic Optimization of Constrained Combinatorial Problems

2.1 Introduction

This chapter introduces a novel optimization approach for discrete multi-objective optimization problems. This efficient optimization is essential for an effective Design Space Exploration (DSE) in the automotive domain. In particular, the optimization problems in the automotive domain are characterized by a huge search space due to the problem size and complexity as well as by stringent constraints due to several domain-specific requirements. In the following, the proposed efficient optimization is introduced and compared on a set of test cases from different domains before it is applied successfully to the DSE of automotive networks in Chapter 3.

Meta-heuristic algorithms are successfully applied to many complex optimization problems. In particular, some of these meta-heuristic optimization algorithms like Evolutionary Algorithms (EAs) perform very well on multi-objective problems. A major shortcoming of these meta-heuristic optimization algorithms is the missing of capability of innately handling arbitrary constraints. Though several generic and specific methods were researched to overcome this drawback, these methods tend to perform badly in case of a general constrained combinatorial problem where the search space is discrete and linearly constrained. Such a *constrained combinatorial problem* is defined as follows:

Definition 2.1 (Constrained Combinatorial Problem)

$$\begin{aligned}
&\text{minimize } f(\mathbf{x})\\
&\text{subject to:}\\
&\quad A\mathbf{x} \leq b \textit{ with } \mathbf{x} \in \{0,1\}^n, A \in \mathbb{Z}^{m,n}, b \in \mathbb{Z}^m
\end{aligned}$$

The objective function f might be multi-dimensional and non-linear. In single-objective optimization, the feasible set of solutions is totally ordered, whereas in multi-objective optimization problems, the feasible set is only partially ordered and, thus, there is generally not only one global optimum, but a set of *Pareto-optimal solutions*. A Pareto-optimal solution is better in at least one objective when compared to any other feasible solution. The search space $X = \{0,1\}^n$ is restricted to binary values, but allows integer values by a binary encoding. The *feasible search space* $X_f \subseteq X$ is restricted by a set of constraints which are subsumed in the stated matrix inequation $A\mathbf{x} \leq b$. Thus, the constraints have to be linear or linearizable.

In case of a relatively small feasible search space, common constraint-handling methods like penalty functions or local repair algorithms are more focused on the search for feasible solutions than on the optimization of the objectives. Figure 2.1 illustrates the shortcomings of a variation of a feasible solution in a constrained search space where the resulting solutions might become infeasible. As a result, only a slow convergence towards the optimal solutions is reached typically. In some cases it might even happen that the meta-heuristic optimization algorithm is not able to find even a single feasible solution. On the other hand, using exact approaches like Integer Linear Programming (ILP) is prohibited by the condition that the objective function is multi-dimensional and non-linear.

To overcome the drawbacks of known optimization methods for the constrained combinatorial problem, a novel approach is proposed in this chapter. This hybrid approach combines the benefits of ILP and meta-heuristic optimization methods, particularly EAs. Since the constraints in Definition 2.1 are restricted to binary variables, a backtracking-based ILP solver might be used to find feasible solutions. This so-called Pseudo-Boolean (PB) solver is incorporated into the meta-heuristic optimization process to enable a constraint handling and preserve the feasibility of the solutions.

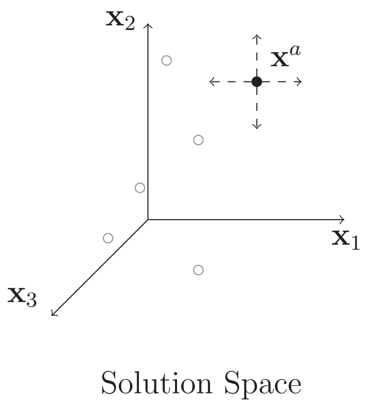

Figure 2.1: Illustration of a common operator that varies a solution $\mathbf{x}^a$ in a constrained search space. The circles correspond to feasible solutions. The variation of solution $\mathbf{x}^a$ may result in infeasible solutions.

Two basically different approaches are proposed that allow integrating a PB solver in the optimization process. First, a decoding approach is presented where the meta-heuristic is used to vary the branching strategy of a PB solver instead of varying the solutions directly [LGHT07]. As a result, the PB solver is used with the branching strategy to obtain feasible solutions. The second approach presents feasibility-preserving operators that are used by the optimization algorithm to vary the solutions inside the feasible search space [LGHT08b, LGT08]. For each operator an individual scheme for the branching strategy for PB solver is proposed. In case of an unconstrained problem, these operators degrade to the known bitwise operators. In particular, this work presents neighborhood, mutation, and crossover operators.

Several test cases give evidence of the benefits of the feasibility-preserving optimization approaches. A random set of single-objective test cases is selected from the PB Evaluation [MR09]. These test cases allow a fair comparison of the feasibility-preserving techniques with a penalty approach based on the convergence towards the optimal objective value. Additionally, two-dimensional optimization problems show the applicability in the multi-objective domain.

An introduction of the meta-heuristic optimization of the constrained combinatorial problem is given as follows:

- An extensive presentation of related work. (cf. Section 2.2)
- A detailed introduction on the PB problem. (cf. Section 2.3)

In summary, this chapter provides the following contributions to the constrained combinatorial problem as defined in Definition 2.1:

- A decoding scheme based on a PB solver that ensures the feasibility of solutions. (cf. Section 2.4)
- An operator scheme based on a PB solver that preserves the feasibility of solutions. (cf. Section 2.5)
- A meaningful set of test cases that gives evidence of the superiority of the proposed methods compared to known constraint handling methods in meta-heuristic optimization. (cf. Section 2.6)

2.2 Related Work

This section discusses existing work related to meta-heuristic optimization and constraint handling methods known from literature. Finally, known hybrid optimization algorithms that combine exact and heuristic algorithms are discussed.

2.2.1 Meta-heuristic Optimization

Meta-heuristic optimization methods are commonly used for complex problems where common optimization techniques like Linear Programming (LP), Quadratic Programming (QP), Geometric Programming (GP), etc. are not applicable due to their restrictive expressiveness. This is often the case if the problem has multiple non-linear objectives or constraints as well as a complex underlying problem representation.

The domain of meta-heuristic optimization comprises methods such as Evolutionary Algorithm (EA) approaches [Bäc96], Simulated Annealing (SA) [KGV83, Čer85], and Particle Swarm Optimization (PSO) [KE95]. The abovementioned meta-heuristic optimization algorithms are inspired by nature and, therefore, also embraced by the term Evolutionary Computation (EC) [Fog95].

Evolutionary Algorithms

An EA is a population-based optimization algorithm inspired by biological evolution. The first EAs were developed independently by different research groups around the world in the 1960s. Lawrence J. Fogel coined the term Evolutionary Programming in 1962 [Fog62, FOW66] for his evolutionary approach to solve prediction models. Also in 1962, John H. Holland initiated the Genetic Algorithm (GA) approaches [Hol62] resulting in a pioneering book published in 1975 [Hol75]. His work was motivated by the development of robust adaptive systems. The work on the Evolution Strategy (ES) started in the 1960s and was further developed in the 1970s by Ingo Rechenberg and Hans-Paul Schwefel [Rec71, BS02]. The ES approaches were used to solve continuous parameter optimization problems. All these approaches share the common idea of using *reproduction* and *natural selection*. Therefore, these approaches are commonly embraced by the term EA. Numerous books have been published on the EA topic such as [Dav91, Mic96, BNKF98].

The main procedure of an EA is based on the reproduction and selection that are performed alternately in an iterative process to optimize a given objective. The reproduction creates new individuals from the current population using the *mutation* and *crossover* operators. These crossover and mutation operators are problem specific. There exist numerous general-purpose crossover and mutation operators for real, integer, binary values and also for problem specific data-structures. The task of the selection is to remove the worst individuals to ensure a convergence of the algorithm towards the optimal solutions. For single-objective optimization this might be a pure elitism selection that always removes the worst individuals or a probabilistic method like the roulette wheel selection.

In recent years, huge efforts were made to adapt the selection to multi-objective problems. The best known and commonly used algorithms for multi-objective selection are the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [ZT99, ZLT02], the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [SD94, DAPM00], and the Indicator Based Evolutionary Algorithm (IBEA) [ZK04].

Simulated Annealing

SA is an optimization algorithm inspired by the annealing process in metallurgy. It is a further development of the Metropolis algorithm [MRR+53] and was developed independently by Kirkpatrick et al. in 1983 [KGV83] and Černỳ in 1985 [Čer85].

The algorithm varies a single solution by a neighborhood operator. The common neighborhood operator for binary problems is a bit flip or the sampling from a normal distribution for real-valued problems. However, also problem-specific neighbor operators were studied and successfully applied to arbitrary problems. The common SA improves a single solution iteratively. A new solution is either accepted or rejected based on a probability value that is calculated using a continuously decreasing temperature function.

The general SA is a single-objective optimization algorithm. However, also multi-objective variants were studied [TK07, BSMD08]. A further extension of the SA is the Tabu Search (TS) [Glo89, Glo90] where each found solution is excluded from the search space to enable a faster convergence towards the optimal solution.

Particle Swarm Optimization

PSO is an optimization algorithm based on swarm intelligence using social-psychological principles. It was first published in 1995 by James Kennedy and Russell C. Eberhart [KE95].

The main procedure is based on particles that are moving in the search space. As a result, the algorithm mainly targets continuous problems. A swarm consists of multiple particles that change their position on each iteration. One single particle is attracted by its local best position and the global best particle. Here, the quality of an particle depends on the objective function. There exist several variations including extensions for multi-objective optimization [CCPL04].

Miscellaneous methods

In addition to the abovementioned meta-heuristic optimization algorithms there exist other approaches which are outlined in the following. Differential Evolution (DE) [SP95] is an optimization approach tailored for continuous search spaces and shares several similarities with PSOs. Ant Colony Optimization (ACO) [DMC96] is restricted to optimize paths through graphs inspired by ant behavior. Meta-heuristic algorithms like Artificial Immune Systems (AIS) [FPAC94], Harmony Search (HS) [GKL01], and many others are tailored for specific optimization problems. Moreover, there exist various hybrid combinations of the discussed algorithms.