Improving Neighborhoods for Local Search Heuristics

Part 2

Peter Brucker
Universität Osnabrück
Johann Hurink
Universität Osnabrück
Frank Werner
Technische Universität Magdeburg

Extended Abstract

Local search techniques are useful tools for solving discrete optimization problems. These methods depend on an underlying neighborhood structure. Usually, the quality of the neighborhood structure has some important influence on the methods.

As mentioned in Part 1 our approach is to replace the original feasible set S_1 of an discrete optimization problem by the subset S_2 of all $s \in S_1$ which are locally optimal with respect to a neighborhood structure $\mathcal{N}_1(s), s \in S_1$ on the set S_1. On the new set S_2 we will define operator sets $OP_2(s), s \in S_2$ which define a new neighborhood structure $\mathcal{N}_2(s), s \in S_2$ on the set S_2.

We apply this approach to the following NP-hard scheduling problems.

(a) $\boldsymbol{P2 \mid\mid C_{max}}$

$P2 \mid\mid C_{max}$ denotes the problem of scheduling n jobs $i = 1, \ldots, n$ with processing times $p_i (i = 1, \ldots, n)$ on two identical parallel machines such that the makespan is minimized.

(b) $\boldsymbol{1 \mid prec \mid \sum C_i}$

$1 \mid prec \mid \sum C_i$ denotes the problem of scheduling n jobs $1, ..., n$ with processing times p_i $(i = 1, ..., n)$ on one machine such that the mean flow time is minimized. Between the jobs precedence relations $\rightarrow$ are given.

(c) $\boldsymbol{1 \mid\mid \sum |T_i}$

$1 \mid\mid \sum |T_i$ denotes the problem of scheduling n jobs $1, \ldots, n$ with processing times p_i $(i = 1, ..., n)$ and due dates d_i $(i = 1, ..., n)$ on a single machine such that the total tardiness $\sum T_i$ is minimized.

For each neighborhood defined in Part 1 for the problems (a), (b) and (c) we will

- characterize the local optima
- define a new neighborhood on the set of the local optima
- prove that the new neighborhood is connected.

(a) The solution set S_1 is given by all possible partitions (I_1, I_2) of the jobs. We define the neighborhood $\mathcal{N}_1$ by the operators *move*(i) which move the job i to the opposite set $(i = 1, \ldots, n)$.

Let (I_1, I_2) be a feasible solution with $s_1 \leq s_2$ (s_i denotes the sum of processing times in I_i). Then (I_1, I_2) is locally optimal with respect to $\mathcal{N}_1$ iff for all $i \in I_2$ we have $p_i < \Delta := s_2 - s_1$.

For the definition of the neighborhood $\mathcal{N}_2$ an operator *localopt*(π) is of crucial importance. *localopt*(π) calculates iteratively for a given solution (I_1, I_2) a corresponding locally optimal solution. In each step *localopt*(π) searches for a job which violates the condition for local optimality. This job will be moved to the opposite set. Therefore in each step the makespan will decrease. If more than two jobs violate the condition for local optimality the given order π of the jobs will determine which of the jobs is moved.

The operator sets $OP_2(s)$, $s \in S_2$ which define the neighborhood $\mathcal{N}_2$ are defined by:

$$OP_2(s) = \{localopt(\pi^*) \circ move(i) \mid i = 1, \ldots, n\},$$

where '$\circ$' denotes the composition of operators and π^* denotes the shortest processing time sequence of the jobs $1, \ldots, n$. These operators first move a job i and afterwards the operator *localopt*(π^*) transforms the new solution again into a locally optimal solution.

With the above choice of π^* connectivity of the secondary neighborhood can be established.

(b) The solution set S_1 is given by all sequences $\pi = (\pi_1, \ldots, \pi_n)$ of the jobs which are compatible with the precedence relations $\rightarrow$. We define the neighborhood $\mathcal{N}_1$ by the operators *exchange*(i) which exchange the jobs π_i and π_{i+1} of the given sequence $(i = 1, \ldots, n-1)$. Due to the Smith rule a sequence π is locally optimal with respect to $\mathcal{N}_1$, iff either $p_{\pi_i} \leq p_{\pi_{i+1}}$ or $\pi_i \rightarrow \pi_{i+1}$ holds for $i = 1, \ldots, n-1$.

For the definition of the neighborhood $\mathcal{N}_2$ we again use an operator *localopt* which calculates iteratively for a given sequence π a corresponding locally optimal solution. In each step *localopt* shifts the job π_i to the left until π_i and its predecessor fulfill the condition of a locally optimal solution for the first time. These shifts are compositions of *exchange* operators which all decrease the mean flow time.

The operators which define $\mathcal{N}_2$ consist of two parts. First a given solution is perturbed by shifting a job to the left or to the right. Afterwards the operator *localopt* is used to calculate a locally optimal solution corresponding to the perturbed solution.

There are two types of shift operators which perturb a solution. The operators *left*(i) will shift the job from position i to a position $j \leq i$, $(i = 2, \ldots, n)$ and the operators

$right(i)$ will shift the job from position i to a position $j \geq i$, $(i = 1, ..., n-1)$. In both cases we try to calculate j in such a way that *localopt* will not reverse the changes of $left(i)$ resp. $right(i)$. Since the operators $left(i)$ and $right(i)$ are compositions of exchange operators; i.e.

$$left(i) = exchange(j) \circ ... \circ exchange(i-1)$$

$$right(i) = exchange(j-1) \circ ... \circ exchange(i),$$

we determine j as the first position such that one of these exchange operators will lead to a decrease of the mean flow time. In this case the operator *localopt* can not reverse the exchanges of $left(i)$ resp. $right(i)$, since *localopt* produces only exchanges with decreasing mean flow time. However not in all cases this will be possible.

For a solution $\pi \in S_2$ the set of operators $OP_2(\pi)$ which define the neighborhood $\mathcal{N}_2$ now are defined by:

$$OP_2(\pi) = \{localopt \circ left(i) \mid localopt \circ left(i)(\pi) \neq \pi;\ i = 2, ..., n\}$$

$$\cup \{localopt \circ right(i) \mid localopt \circ right(i)(\pi) \neq \pi;\ i = 1, ..., n-1\}.$$

It is possible to prove that the neighborhood $\mathcal{N}_2$ is connected.

(c) The solution set S_1 is given by all sequences $\pi = (\pi_1, ..., \pi_n)$ of the jobs which are compatible with the precedence relations $\rightarrow$ defined in Part 1. As in (b) the neighborhood $\mathcal{N}_1$ is defined by the operators $exchange(i)$ $(i = 1, ..., n)$.

In contrast to (b) it depends not only on the exchanged jobs π_i and π_{i+1} but also on the start time of job π_i whether or not the operator $exchange(i)$ will decrease the objective value. We introduce the notation $(i,k)_T$ in such a manner that the sequence i,k is preferred to the sequence k,i when the first job starts at time T. This means $(i,k)_T$ is sufficient for the fact that interchanging the adjacent jobs i and k in a sequence π where the first job i starts at time T does not lead to an improvement of the objective value.

Let i,k be two jobs with $i < k$. Then we define

$$(i,k)_T \text{ iff } p_i \leq p_k \text{ or } p_i > p_k \text{ and } T \leq d_k - p_i$$

and

$$(k,i)_T \text{ iff } p_i > p_k \text{ and } T > d_k - p_i.$$

If the exchange of two jobs will not change the objective value, this definition gives a rule which of the two jobs should be scheduled first.

We now define that a feasible sequence $\pi \in S_1$ belongs to the set S_2 of locally optimal sequences iff $(\pi_i, \pi_{i+1})_{S(i)}$ holds for all $i = 1, \ldots, n-1$ where $S(i)$ is the starting time of the job on position i in π. According to the definition of $(i,k)_T$ several locally optimal sequences with the same objective value which are connected in the neighborhood $\mathcal{N}_1$ are represented by one of these sequences in S_2.

In order to define a neighborhood structure on the set S_2 we again use an operator *localopt* which calculates a locally optimal sequence. However this operator is not comparable with the *localopt* operators for the two problems considered before since it will start with a subsequence of jobs and not with a complete sequence $\pi \in S_2$.

We call a subsequence $\pi^F = (\pi_1^F, ..., \pi_k^F)$, $k \leq n$ a **final sequence** if the jobs π_i^F and π_{i+1}^F fulfill the condition of local optimality when the job π_i^F starts at time

$$\sum_{j=1}^{n} p_i - \sum_{j=i}^{k} p_{\pi_j^F},$$

i.e. if the jobs of π^F are scheduled at the end of a complete sequence.

Starting with a final sequence π^F *localopt* constructs a locally optimal schedule $\pi' \in S_2$ with π^F as final sequence if such a sequence exists. However, there exist partial sequences which cannot be completed to a sequence $\pi' \in S_2$.

The operator *localopt* consists of two operators. The first operator $localopt1(R)$ constructs for a given set R of jobs a locally optimal schedule π^R, where the first job of π^R starts at time 0.

If we apply this operator for a given final sequence π^F to the set R of unscheduled jobs, i.e. to the set of jobs not contained in π^F, the concatenation of π^R with the final sequence π^F does not necessarily lead to a locally optimal sequence $\pi' = (\pi^R, \pi^F)$ because the last job of π^R and the first job of π^F may violate the condition of local optimality. In this case we try to extend π^F to a final sequence (π^M, π^F) such that the set R of still unscheduled jobs (jobs not contained in π^M or π^R) can be scheduled with $localopt1(R)$ and the concatenation of π^R and (π^M, π^F) leads to a locally optimal sequence. The corresponding operator will be denoted by *localopt*2. If it is not possible to extend π^F to a locally optimal sequence, *localopt*2 will stops with this information.

Summarizing, we first apply *localopt*2 to a final sequence π^F and extend π^F to a final sequence (π^M, π^F) (if this is possible). Afterwards we apply *localopt*1 to the set R of still unscheduled jobs and we get a locally optimal sequence $\pi' = (\pi^R, \pi^M, \pi^F)$. This yields:

$$localopt(\pi^F) := localopt1(R) \circ localopt2(\pi^F).$$

Again the operators which define $\mathcal{N}_2$ consist of two parts. First a given sequence π is perturbed by shifting a job to the right. We define a shift operator $right(i,j)$ which shifts a job from position i, say job k, to position $j > i$. For a given value i we allow only operators $right(i,j)$ where

- $(k, \pi_{j+1})_S$ holds (S denotes the starting time of job k)
- no precedence relation $k \rightarrow \pi_u$ $(i+1 \leq u \leq j)$ exists.

Afterwards the operator *localopt* is applied to the final sequence $\pi^F = (k, \pi_{j+1}, \ldots, \pi_n)$ to obtain again a locally optimal solution.

For a solution $\pi \in S_2$ the set of operators $OP_2(\pi)$ now is defined by:

$$OP_2(\pi) = \{localopt(\pi^F) \circ right(i,j)(\pi) | i = 1, \ldots, n-1; \; right(i,j) \text{ is defined}; \; \pi^F \text{ can be completed by } localopt(\pi^F)\}$$

Again it is possible to prove that the neighborhood defined by $OP_2(\pi)$ is connected.